TALRIK JUNIOR PROFESSIONAL™
TJ PRO™
USERS MANUAL

by
Keith L. Doty

Gainesville, Florida   www.mekatronix.com   Tech Questions: tech@mekatronix.com
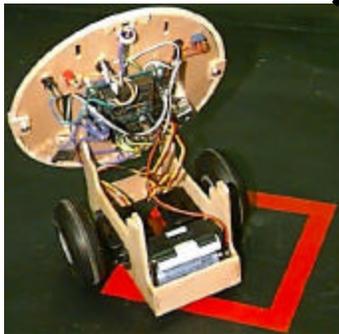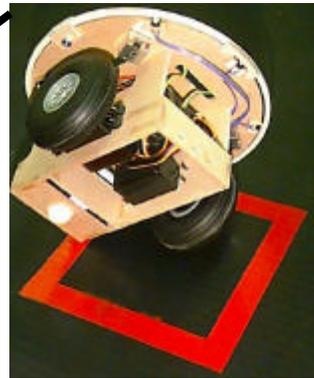
AGREEMENT

This is a legal agreement between you, the end user, and Mekatronix™. If you do not agree to the terms of this Agreement, please promptly return the purchased product for a full refund.

1. **Copyright Notice.** Mekatronix™ hereby grants to any individuals or organizations permission to reproduce and distribute copies of this document, in whole or in part, for any personal or non-commercial educational use only. This copyright notice must accompany any copy that is distributed.

2. **Copy Restrictions.** Other than cases mentioned in **Copyright Notice**, no part of any Mekatronix™ document may be reproduced in any form without written permission of Mekatronix™. For example, Mekatronix™ does not grant the right to make derivative works based on these documents without written consent.

3. **Software License.** Mekatronix™ software is licensed and not sold. Software documentation is licensed to you by Mekatronix™, the licensor and a corporation under the laws of Florida. Mekatronix™ does not assume and shall have no obligation or liability to you under this license agreement. You own the diskettes on which the software is recorded but Mekatronix™ retains title to its own software. The user may not rent, lease, loan, sell, distribute Mekatronix™ software, or create derivative works for rent, lease, loan, sell, or distribution without a contractual agreement with Mekatronix™.

4. **Limited Warranty**. Mekatronix™ strives to make high quality products that function as described. However, Mekatronix™ does not warrant, explicitly or implied, nor assume liability for, any use or applications of its products. In particular, Mekatronix™ products are not qualified to assume critical roles where human or animal life may be involved. For unassembled kits, the user accepts all responsibility for the proper functioning of the kit. Mekatronix™ is not liable for, or anything resulting from, improper assembly of its products, acts of God, abuse, misuses, improper or abnormal usage, faulty installation, improper maintenance, lightning or other incidence of excess voltage, or exposure to the elements. Mekatronix™ is not responsible, or liable for, indirect, special, or consequential damages arising out of, or in connection with, the use or performances of its product or other damages with respect to loss of property, loss of revenues or profit or costs of removal, installation or re-installations. You agree and certify that you accept all liability and responsibility that the products, both hardware and software and any other technical information you obtain has been obtained legally according to the laws of Florida, the United States and your country. Your acceptance of the products purchased from Mekatronix™ will be construed as agreeing to these terms.

MANIFESTO

 Mekatronix™ espouses the view that personal autonomous physical agents will usher in a whole new industry, much like the personal computer industry before it, if modeled on the same beginning principles:

- Low cost,
- Wide availability,
- Open architecture,
- An open, enthusiastic, dynamic community of users sharing information.

Our corporate goal is to help create this new, exciting industry!

WEB SITE: http://www.mekatronix.com
Address technical questions to tech@mekatronix.com
Address purchases and ordering information to an authorized Mekatronix Distributor
http://www.mekatronix.com/distributors

Disclaimer

While Mekatronix™ has placed considerable effort into making these instructions accurate, Mekatronix™ does not warrant the results and the user assumes the risks to equipment and person that are involved.

TABLE of CONTENTS

Gainesville, Florida   www.mekatronix.com   Tech Questions: tech@mekatronix.com

LIST of FIGURES

LIST of TABLES

Gainesville, Florida   www.mekatronix.com   Tech Questions: tech@mekatronix.com

1. SAFETY AND HANDLING OF TALRIK JUNIOR PROFESSIONAL™ (TJ PRO™)

1.1 TJ PRO™'s Static Sensitive Parts

Some of TJ PRO™'s components are static sensitive. These are located on the printed circuit boards. Do not touch these boards without being properly grounded. Static discharge can destroy these parts.

1.2 Caution for Biological Organisms

Most individuals find TJ PRO™, exciting, enjoyable, amusing, and fascinating. TJ PRO™'s small size minimizes its[1] threat to organic life, although pets and small children might find TJ PRO™ either an interesting playmate or an alien to avoid. A child's hands, feet, and mouth probing TJ PRO™ may lead to minor injury to both child and robot, so, to eliminate adverse reactions with pets, children, or nervous adults, be sure TJ PRO™ operates under supervision by someone who knows how to turn it (him!) off. Remember, TJ PRO™ is an autonomous machine and, as such, becomes "out of control" once set in motion. An add-on IR remote control package can be purchase separately to allow you to override TJ PRO™'s autonomous mode.

1.3 Holding, Carrying, and Transporting TJ PRO™

TJ PRO™'s small size and portability make it ideal for safe demonstrations and presentations of machine intelligence algorithms. Porting TJ PRO™ from desk-top platform to floor and back, the most frequent carrying operation, requires handling TJ PRO™ carefully. The recommended carrying technique is to firmly hold TJ PRO™ from the rear with the thumb on top of the key holding the plate down and the battery case floor and tail skid resting on the remaining four fingers. Users at their own risk may devise other carrying techniques.

To transport TJ PRO™ for long distances, place it securely padded into a suitably sized box.[2] Do not expose TJ PRO™ to extreme cold or heat in an automobile during winter and summer for more than a few hours. Although TJ PRO™ has survived a hot automobile for 12 hours in *40° C*, such treatment cannot help but reduce its lifetime.

TJ PRO™ has flown in airline luggage compartments across the Atlantic, ridden in automobiles, passed around in a class of electrical and computer engineering students, and used by middle school students for schools science projects, all without ill effects! The following caution, however, is our official advisement.

---

[1] We want to say "his", since TJ PRO is definitely a male name for us. Some future robots will certainly bear female names. We will resist "his" for now, but the anthropomorphic tendency to use personal pronouns in referring to TJ PRO is hard to resist and we cannot guarantee that personal pronouns usage will not appear!
[2] An accessory TJ PRO™ carrying case will be offered soon.

Gainesville, Florida   www.mekatronix.com   Tech Questions: tech@mekatronix.com

Caution: *TJ PRO™ should not be exposed to moisture, or continuous high humidity (>90%), or high temperatures, 40° C (100°F), or low temperatures, 5°C (40°F).*

1.4 Storage

Place TJ PRO™ in a labeled (don't forget where "he" is), enclosed, sealed, padded box in an office or home environment when storing for extended periods of time.

2. TJ PRO™ SET-UP

You may find that the TJ PRO™ Assembly Manual provides useful nomenclature and diagrams depicting the various components and structures and you may need to refer to it from time to time.

2.1 Getting TJ PRO™ Ready

This section tells you how to unpack and prepare TJ PRO™ for operation. If you bought a TJ PRO™ kit you will have already read the TJ PRO™ Assembly Manual and put it together and you can skip the next section.

2.2 Unpacking an Assembled TJ PRO™

1. Carefully unwrap TJ PRO™ from the bubble wrap being careful not to catch wires or drop the robot.
2. Visually inspect the robot for any obvious damage.
3. Make sure that no cables have become disconnected from the robot during shipment. If cables have disconnected, it is usually obvious how to reconnect them as they were.
4. In the case of the IR detectors, the black side of the cable should line up with the edge of the case. Check to insure the IR connectors are fully plugged into the IR emitters ("eyes" on top of the robot plate). The IR emitters are diodes, so if the plugs are not oriented properly, the IR emitters will not emit. Refer to the TJ PRO™ Assembly Manual if in doubt about the connections.
5. Check the bumper to see if it is pushing up against any switches. If it is, just gently shift the bumper away from the switch. Latter program testing will reveal any specific problems.
6. Download all software and documentation from the vendor's web site, unless you made special arrangements to purchase diskette or CD versions. Ask your vendor how to do the download.

7. You may have also purchased the following (recommended)
   - A wall plug adapter for recharging the robot ,
   - An MB2325 serial communications board and cable,
   - 6 Nickel Cadmium rechargeable batteries.

2.3 Computer Requirements

Program development for the TJ PRO™ can be performed on a low cost DOS system. Nothing more is required. Users with more expensive, sophisticated computer systems, of course, can run DOS under the *Windows95* environment.

2.3.1 Basic system (DOS)

The following computer system will meet all the basic hardware support for developing TJ PRO™ programs and applications.

1. 286-PC or later processor running under *DOS* with several Megs of available hard disk space and 1 Meg of RAM.
2. An available serial port. We will assume that COM1 is used. See the next section about COM port problems to avoid.
3. 1.44 Meg floppy

2.3.2 A newer system (Windows95)

The following high-end hardware offers a more sophisticated and easier to use working environment.

1. A high speed (100MHz or better) computer running *Windows 95*
2. **NO** modem (see the section titled *COM port problems to avoid*, page 10)
3. A mouse on COM2
4. A printer
5. Access to the *Internet*, especially the *World Wide Web*,
6. *Windows* terminal (*terminal.exe*)

2.4 COM port problems to avoid

Randy Sargent's Interactive C (*IC*) and Motorola's *Pcbug11* , both freeware, use the serial port in a manner that does not permit sharing the serial interrupt. Therefore, the port that communicates with the robot should not share the same interrupt with any other system resource. For example, COM1 and COM3 share the same interrupt in DOS, as do COM2 and COM4. So, if the robot is connected to COM1, nothing else should be connected to COM1 or COM3 or use their interrupts. We have seen some extreme instances when the internal modem of a computer had to be removed to get *IC DOS* version or *Pcbug11* running, because the modem used the same interrupt as the robot COM port and disrupted the robot serial communication.

2.5 Serial Communication between Host Computer and TJ PRO™

This section assumes TJ PRO™ has fully charged batteries and the processor functions. Without either of these prerequisites, the user cannot begin.

The MB2325 serial communications board generates the proper voltage conversions that permit the user to download and upload code and data to TJ PRO™ via a 6-wire serial communications link. Whenever testing programs on TJ PRO™, the user will connect the host computer with TJ

PRO™ via this interface. Rather than duplicate this interface circuit on each robot, Mekatronix™ chose to make a single external interface for the host computer. This approach has a twofold advantage, 1) removal of this circuitry from the robot platform saves battery energy and reduces printed circuit board cost, 2) only one MB2325 board is required per computer instead of one per robot. In multiple robot systems, or swarm systems, this approach yields an increasing cost advantage.

2.6 Plugging MB2325 into the host computer

The MB2325 serial communications board (Figure 3, page 18) has a 25-pin, female, type D-connector. On most computers COM1 is a 25-pin male D-connector. If your computer has a 9-pin connector you will need a 9- to 25-pin converter (available at most computer and electronics stores). When the MB2325 is plugged into the COM1 port of the host computer D1 will light. This light lets you know that the serial port on the host computer is working. If D1 does not light, something is wrong with the serial port of the host computer and the robot will not be able to communicate with it.

2.7 Connecting the Host computer to TJ PRO™

After the MB2325 is connected to the computer, take the 6-pin rainbow colored cable and insert either end onto the only 6-pin male header (J2) on the MB2325.   On the back of the robot is a 6-pin rainbow cable with a male connector. Connect this male header to the female connector on the other end of the cable coming from the computer. This end is also a keyed connector. Now, place the *DOWN-LOAD / RUN* switch to the *DOWN-LOAD* position. Turn the *ON/OFF* switch to *ON*. At this point the red LED on the robot's top board should light. Press and momentarily hold *RESET* (the red button) on the robot. With *RESET* pressed, D2 on the MB2325 should light. If D2 does not come on, reverse the 6-wire connector at one end only and try again. D2 should now light with *RESET* pressed. When *RESET* is released, D2 on the MB2325 turns off.

2.7.1 Verify proper operation

The lights on the MB2325 let you know that the basic connections are working. If the POWER-ON light does not light when the power switch is turned to *ON*, the robot may not be charged. If the batteries are charged, check battery and power connections to the circuit boards. Particularly, check underneath the robot to verify the battery cable has not come undone from the power connector. If the lights on the MB2325 board do not function as specified, check that no other programs or devices are using the serial port.

2.8 Software Language Support

TJ PRO™ programs can be developed for the MC68HC11 with Interactive C (IC), a C interpreter, Image Craft C (ICC11), the compiled version of C, or with Motorola MC68HC11 Assembly Language. Both ICC11 and IC come in a DOS version and a WINDOWS version. Mekatronix™ sells both versions of ICC11 and only the WINDOWS version of IC. The DOS version of IC and the Motorola MC68HC11 Assembly Language are freeware and can be downloaded free from the Mekatronix web site.

IMPORTANT!

Gainesville, Florida   www.mekatronix.com   Tech Questions: tech@mekatronix.com

IC for WIN95 and ICC11 require special library files for the TJ PRO™ supplied only by Mekatronix and not by any other vendor. To purchase IC for WIN95 or ICC11 contact one of Mekatronix authorized distributors listed at http://www.mekatronix.com/distributor *to insure you get the necessary library files with your purchase.*

IC and ICC11 permit extensive program development on TJ PRO™ using the Standard C language. IC allows you to write compact code that takes a bit longer to execute, often not a problem with robots, while ICC11 provides a developers environment and faster execution of code.

2.9 PROGO™ Language

Mekatronix developed the robot language PROGO™ to simplify the writing of programs for its robot product line. The language can be learned in several hours and used immediately. PROGO™ was designed with novice programmers in mind and yet experts can appreciate its conveniences. For one, PROGO™ maintains compatibility with the C Language in that C statements can be mixed with PROGO™ statements without syntactic error. For example, PROGO™ supports only integer data types directly, but any C data type can be defined and used in a PROGO™ program without modification. I generally program in PROGO™ and simply use C definitions or statements when required, which is seldom!

PROGO™ operates under three systems: IC DOS (freeware), IC for WIN95, ICC11 DOS and WINDOWS versions. Check with a Mekatronix distributor for ordering the PROGO™ Language Reference Manual with PROGO™ sample application programs, both object and source code.

Middle school students have used PROGO™ with great success.

Mekatronix provides technical support for PROGO™.

3. TJ PRO™ OVERVIEW

This section provides the user general orientation to TJ PRO™ .

3.1 What Can TJ PRO™ DO?

TJ PRO™ provides an entry-level platform for the development and testing of machine intelligence and autonomous behavior algorithms. The creative and entertainment value of developing your own intelligent, physically embodied autonomous agent provides tremendous motivation and pleasure. As with most high-tech tools (toys!) the implications have broad scope. By attaching different mechanisms to such an autonomous agent, the user can develop autonomous vacuum cleaners, autonomous lawnmowers, autonomous construction vehicles, autonomous butlers or maids, intelligent kinesthetic art, tile laying robots, security robots, maintenance robots, space station robots, planetoid explorers, undersea mining, rescue vehicles, war robots, and so on. As the community of autonomous robot developers increases, the applications coming forth will be legion, many of which will be totally surprising.

The modest beginning represented by TJ PRO™ and other Mekatronix robots to follow will lay the groundwork and develop the technical infrastructure for a new mechanical species to serve mankind. The arrival of this new species will tremendously affect social, political, religious, philosophical, scientific, engineering, and mathematical interests. To our children's children, these days will seem primitive indeed.

3.2 Operating Environment for TJ PRO™

TJ PRO™ is designed to operate in an office or home environment with smooth tiled floors. Although TJ PRO™ can handle short pile rugs, he spends more energy overcoming rug friction than tile friction and his operational lifetime on a battery charge reduces considerably. Shag rugs and extremely rough surfaces are inimical to TJ PRO™ and should be avoided.

In general, operating environments comfortable to lightly clothed humans will probably be suitable for TJ PRO™. However, we reiterate the previous caution on environmental constraints.

Caution: *TJ PRO™ should not be exposed to moisture, or continuous high humidity (90%), or high temperatures, 40° C (100°F), or low temperatures, 5°C (40°F).*

3.3 Physical Orientation

TJ PRO™'s wheel axis determines the robot's left-to-right axis. The diameter perpendicular to the wheel axis determines the front-to-back axis. The "TALRIK JUNIOR PROFESSIONAL™" label appears on the front of the robot. Figure 1 presents an elevation view of TJ PRO™.
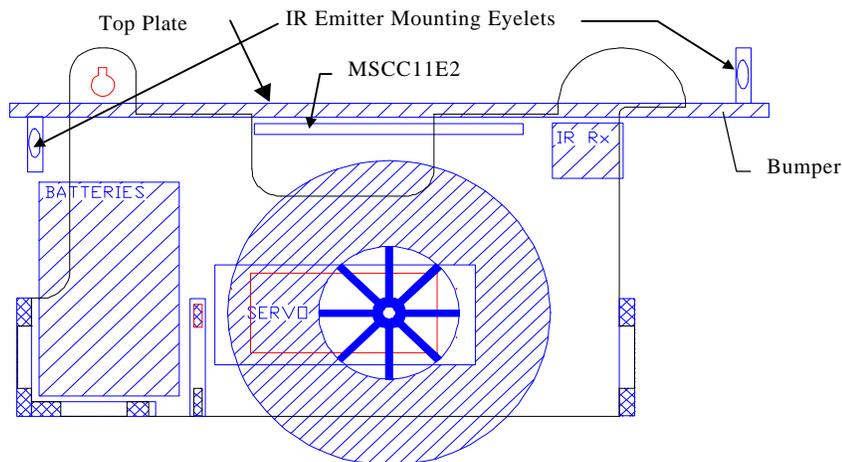
Figure 1 Schematic side view of TJ PR.™

3.4 Sensor Layout

Figure 2 schematizes the standard layout of TJ PRO™'s sensor suite. Table 1 defines the meaning of the labels and a typical application of the sensor. The user is not limited to these applications

13

Gainesville, Florida   www.mekatronix.com   Tech Questions: tech@mekatronix.com

and can devise and implement other schemes, both in layout and in function. The TJ PRO-ARGOS™ extension expands TJ PRO™'s sensory capability by 9 photoresistors, 2 side-looking IR detectors and emitters, and 1 rear-looking digital IR detector for IR communications or remote control.

Table 1 TJ PRO™'s Minimum Sensor Suite

Label	Name	Function
IRDLF	Infrared Detector, Left Front	Proximity Sensor
IRDRF	Infrared Detector, Right Front	Proximity Sensor
BSWLF	Front Bumper Switch, Left Front	Front contact Sense
BSWMF	Front Bumper Switch, Middle Front	Front contact Sense
BSWRF	Front Bumper Switch, Right Front	Front contact Sense
BSWB	Rear Bumper Switch	Rear contact Sense
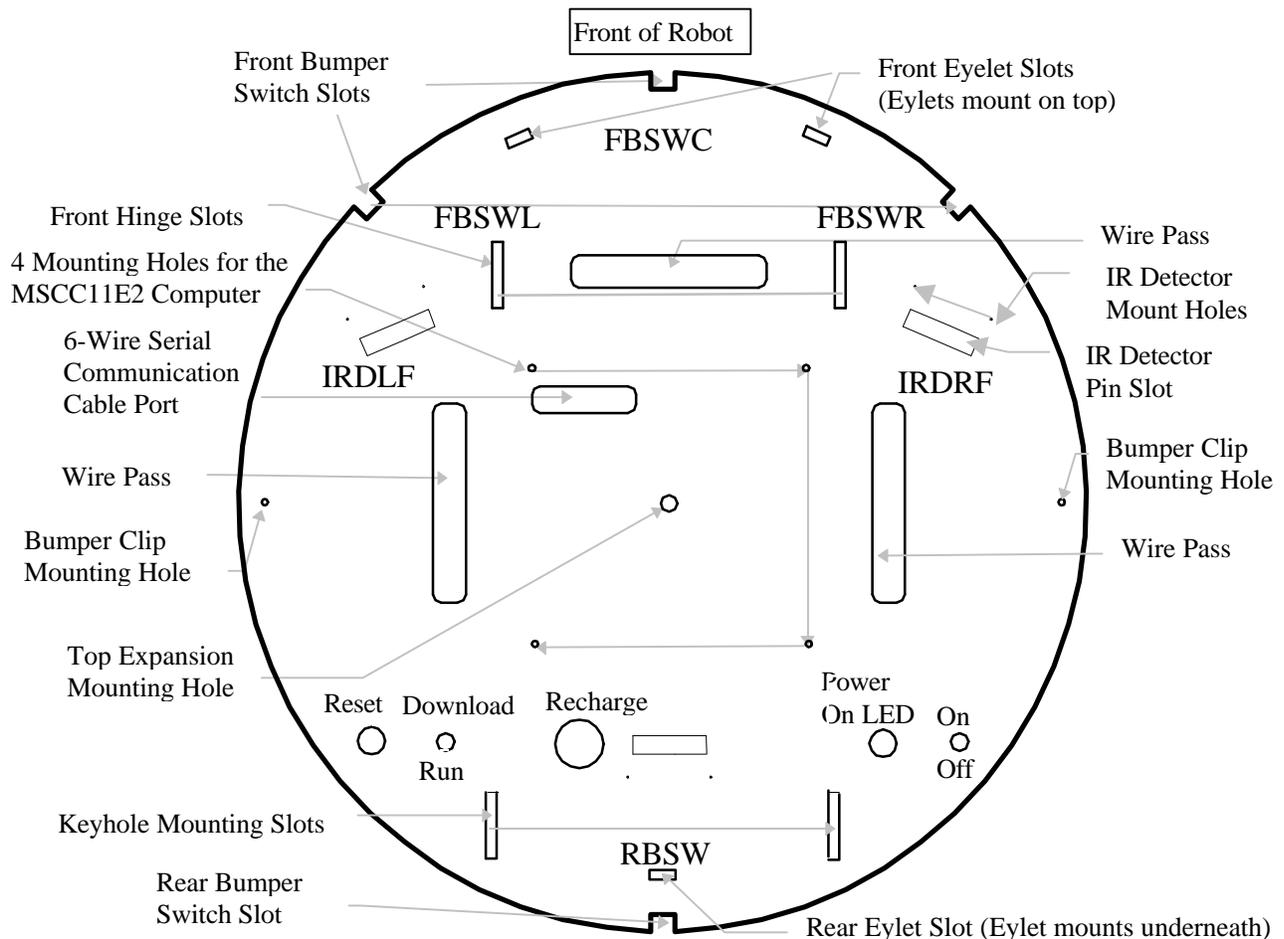
Figure 2 Sensor layout on top of the TJ PRO™ plate. The lead symbols are B= Bumper, IRD = Infrared Detector. The following symbols mean: B = Back, F= Front, SW = Switch, L= Left, M= Middle, R= Right. For example,

BSWB = Bumper-Switch-Back and IRDFL = IR Detector Front Left. The IR emitters and the IR Detectors are typically mounted on opposite sides of the top plate.

3.5 Switches

TJ PRO™ control switches on the top plate near the back provide easy access. The layout of the switches can be determined from Figure 2.

The red *RESET* button resets the MC68HC11E2 microcomputer. The *ON/OFF* switch provides power. The visible red led adjacent to the *ON/OFF* switch lights when the switch is moved to the *ON* position.

In *DOWN-LOAD* mode the *DOWN-LOAD/RUN* mode switch permits the user to download programs using PCBUG11. In *RUN* mode, a press of the reset button will invoke the start-up program the user has loaded into TJ PRO™.

3.6 Batteries

TJ PRO™ requires a minimum of six AA Nickel-Cadmium rechargeable batteries. *Only use nickel-cadmium AA batteries. Alkalines, for example. will produce too much voltage and may damage the electronics.*

The battery pack is located in front of the rear caster on the underneath side of the robot.
TJ PRO™ can run autonomously on a smooth tile floor for about 1.5 to 2 hours on a fully charged battery pack. This time depends critically upon how frequently the motors change speed and the running time of the motors.

NiCad batteries sustain useful power until just before they become fully discharged. So, if you see TJ PRO™ really slowing down, it is time to recharge!

3.7 Recharger

The charging technique is known as a trickle charge. This means that a small current is constantly applied to the batteries. Because this is a low level current the robot can be left charging indefinitely. While the robot is not moving around plug it into the charger. This will keep the batteries charged. Be sure to use only a Mekatronix approved AC adapter.

TJ PRO™'s charger must be at 12 volts and be able to supply 200-500 milliamps. Lower voltages will not charge the batteries and higher voltages my damage the electronics. The charger plugs into the receptacle mounted on the left bridge support.

Recommendation: During program development and high usage times, keep TJ PRO™ connected to the charger whenever possible. This will insure fresh batteries during experimentation.

Caution: If the output DC voltage of the AC adapter exceeds 12 volts, the charging resistors will act like a fuse and burn out. The smoke and smell are unpleasant, so be careful.

3.8 Installing TJ PRO™'s batteries (IMPORTANT: NiCads only)

1. Turn *On/Off* switch to *Off*.
2. Insert six NiCad AA batteries into the pack, oriented with proper voltage polarity as labeled on the pack. Failure to do so may destroy all the electronics.
3. Turn and slide the plate key out of the rear supports and rotate TJ PRO™'s plate forward, exposing the rear battery compartment.
4. Carefully place the battery pack into the battery compartment. Practically any orientation of the pack will work, except with the battery snap terminals facing down!
5. Snap the 9v snap connector to the battery pack and the female connector on the other end into the male power header on the underneath side of the plate.
6. Rotate the top plate down onto the sides, being careful not to pinch any wires between the top plate and the sides.
7. With the top plate down, slide the key into the key slot, turn 90 degrees to lock the key into place.

3.9 Replacing TJ PRO™'s batteries (IMPORTANT: NiCads only)

1. Turn *On/Off* switch to *Off*.
2. Turn and slide the plate key out of the rear supports and rotate TJ PRO™'s plate forward, exposing the rear battery compartment.
3. Unplug the battery connector from the male power header attached to the underside of the plate.
4. Carefully remove the battery pack from the battery compartment.
5. Remove the old batteries and insert six NiCad AA batteries into the pack, oriented with proper voltage polarity as labeled on the pack. Failure to do so may destroy all the electronics.
6. Snap the 9v snap connector to the battery pack (if not already snapped on) and the female connector on the other end into the male power header on the underneath side of the plate.
7. Rotate the top plate down onto the sides, being careful not to pinch any wires between the top plate and the sides.
8. With the top plate down, slide the key into the key slot, turn 90 degrees to lock the key into place.

TJ PRO™ will now have power when the *On/Off* switch is turned to *On.*

3.10 TJ PRO™ Bumper

TJ PRO™'s bumper girdles the plate. Objects higher or lower than TJ PRO™ platform will not be seen. Such obstacles do exist in home and office, and can stall the robot when it strikes them. A stalled robot is not a pleasant thing to watch and hear! The wheels struggle to turn and the motors might eventually overheat and burn out if the surface has high friction. Our recommendation is either to 1) TJ PRO™ proof such obstacles or 2) Hang around and get him out of trouble or 3) Write a program that detects motion, perhaps using the underneath

Gainesville, Florida　　　　www.mekatronix.com　Tech Questions: tech@mekatronix.com

photoresistors of the ARGOS extension as vision flow detector. We recommend 3). The idea behind 3) is that if TJ PRO™ does not detect any forward motion, he assumes he is stuck and backs up, turns around, and leaves the area.

Behind the bumper, four push button switches surround TJ PRO™'s waist. TJ PRO™ is configured only to distinguishes between front and back collisions. Experts can make hardware and software changes to enable TJ PRO™ to distinguish any subset of button closures activated by the bumper.

Caution: DO NOT LIFT or CARRY TJ PRO™ by the bumper.

3.11 Serial Communication

The MB2325 serial communications board (Figure 3) permits the user to download and upload code and data to TJ PRO™ via a 6-wire serial communications link. The 6-wire communications line connects into the MB2325 bidirectional serial communications board at one end (lower-right corner in (Figure 3) and to TJ PRO™'s serial interface at the other (Figure 4). The MB2325 D-connector plugs directly into a COM port of your personal computer 25 pin D-connector or through a serial cable. If your computer serial connector only has a 9 pin D-connector, you will need to acquire a 25 to 9 pin plug converter.

25-pin D-Connector to Host Computer

6-Pin Serial Connection to TALRIK™

Diode D1
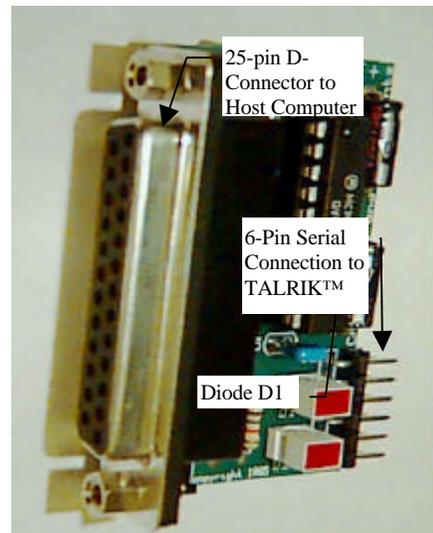
Figure 3 MB2325 Communications Board
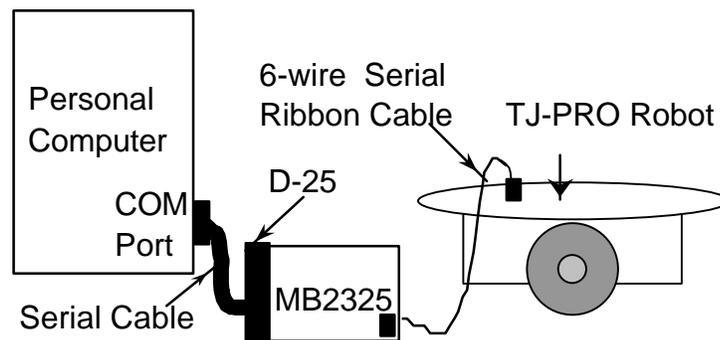
Figure 4. This diagram illustrates the serial connection between a personal computer, the communications board (MB2325) and a Talrik Junior Professional.

3.12 Where is TJ PRO™ During Program Development?

You will probably spend considerable time developing and testing programs to run on TJ PRO™. During the initial phases of this process the user will undoubtedly make many changes and run many variations of the programs. This procedure can be expedited by mounting TJ PRO™ on a test platform that elevates the drive wheels off the floor or desktop so that the wheels turn freely without moving the robot. To maintain fresh batteries during these long development sessions, keep TJ PRO™ connected to the charger. Also, keep the serial communications cable to TJ PRO™ connected to minimize plugging and unplugging it. When the user is ready to test TJ PRO™ on the floor, disconnect both the charger and the serial communications cable from TJ PRO™.

3.13 Can you Physically Damage a TJ PRO™ with Software?

The answer is YES! A program that commands the robot to oscillate between 100% forward and 100% backward for 30 seconds or more will overheat, and possibly burn out, the motor driver chip. Anytime your program controls servos and motors, you should take extra care to insure this type of behavior does not occur.

WARNING!

Rapid oscillations of motor direction and speed for extended periods of time will burn out the motor control chip.

There is another case where software can damage the TJ PRO™. Interactive-C (IC) uses the MC68HC11 "output compare four" (OC4) pin to control its multi-tasking timing. We use OC4 to control a servo in ICC11. If a servo is connected to OC4 and you switch from ICC11 to IC, IC will destroy the attached servo. The rapid changes of OC4 will actually burn out any servo attached to OC4 header.

WARNING!

IC code thrashes OC4 and could easily destroy an attached servo in a short amount of time. Any servo connected to OC4 must be disconnected before running IC code.

3.13.1 Controlling TJ PRO™ Motors

When the batteries are low, the robot may stop after making a series of rapid moves. This behavior usually indicates that the robot has reset because of high motor current surges have drawn the battery voltage down so low that the voltage regulator on the processor drops out. For

example, if your program changes both wheel motors simultaneously from full speed forward to full speed backward, there might be a chance of reset when the batteries are low. These reset-type conditions can be avoided by more careful motor control. Change motor speeds, hence, currents, gradually. What is "gradual" to the motor, however, will hardly be observable to you if done in tens of milliseconds. When you program smooth motor control, the robot motion becomes smooth and looks more "organic".

3.14 Halting a Moving TJ PRO™

Chasing down a moving TJ PRO™ takes more skill than transporting it. A recommended procedure is to chase it from behind and reach down to turn the *ON/OFF* switch *OFF*. You can also program the robot to stop on certain conditions that can be artificially generated. For example, you can program the robot so that a touch of the rear bumper while the robot is going forward will stop it. With the IR remote control unit accessory unit, TJ PRO™ can be stopped by the IR remote.

4. IF YOU DO NOT HAVE A C COMPILER

If you did not purchase ICC11 or the commercial version of IC and you do not have your own C compiler for the MC68HC11 family of computers, you have several options:

1. You can program in the non-commercial (freeware) version of IC, IC DOS, or
2. You can program in MC68HC11 Assembly language and assemble your programs from the Motorola freeware Assembler

The freeware version of IC and the MC68HC11 Assembler can be downloaded from www.mekatronix/downloads. An IC Users Manual can also be downloaded from the Mekatronix web site.

NOTE: Mekatronix does not support these programs and recommends the commercial versions for higher reliability, expanded function, and technical support.

 The Motorola Assembler will generate S19 files from your assembly language programs. S19 files are ASCII files that the Motorola boot loader can read and convert to MC68HC11 machine code. The download process for S19 files is described in Section 14.5 of this manual.

Assembly language programming is beyond the scope of this manual. A good, readable beginning textbook for learning MC68HC11 assembly language programming is:

Microcomputer Engineering, Gene H. Miller, Prentice Hall, 1993, ISBN 0-13-584475-4

5. TJ PRO™ PROGRAMMING SET-UP

The following sections instruct you on how to setup and use your TJ PRO™ Distribution Software package and integrate it with the compiler systems ICC11 WIN95 or ICC11 DOS or the interpreter systems IC WIN95, or IC DOS.

5.1 Computer Requirements

Program development for the TJ PRO™ can be performed either a WINDOWS or DOS system. In a windows system you have the option of running DOS, but not vice-versa. In either case you will find the Mekatronix *High Speed Serial Downloader* (HSSDL11) a useful tool. The HSSDL11 downloads programs into the robot microcontroller at a blistering 115.2KBaud.

5.1.1 Primitive System (DOS)

ALERT!

MICROSOFT™ has announced that DOS will be phased out within a few years. This may affect your choice of WINDOWS or DOS software. In particular, when you upgrade your computer system in the future, you may have compatibility problems if you run your robots from DOS. Mekatronix™ recommends the WINDOWS system for longer term utility!

The following computer system will meet all the basic hardware support for developing TJ PRO™ programs and applications.

1. 286-PC or later processor running under *DOS* with several Megs of available hard disk space and 1 Meg of RAM.
2. An available serial port. We will assume that COM1 is used. See the next section about COM port problems to avoid.
3. 1.44 Meg floppy
4. Optional, but very useful, a terminal program such as *Kermit* for DOS.

5.1.2 Operating from a WINDOWS System (Recommended)

The following high-end hardware offers a more sophisticated, easier to use, more productive working environment.

1. A high speed (100MHz or better) computer running *Windows 95*
7. **NO** modem (see the section titled *COM port problems to avoid*, page 10)
8. A mouse on COM2
9. A printer
10. Access to the *Internet*, especially the *World Wide Web*,
11. *Windows* terminal (*terminal.exe*)

5.1.3 Com port problems to avoid

Randy Sargent's Interactive C for DOS (*IC-DOS*) and Motorola's *Pcbug11* , both freeware, use the serial port in a manner that does not permit sharing the serial interrupt. Therefore, the port that communicates with the robot should not share the same interrupt with any other system resource. For example, COM1 and COM3 share the same interrupt in DOS, as do COM2 and COM4. Therefore, if the robot is connected to COM1, nothing else should be connected to COM1 or COM3 or use their interrupts. We have seen some extreme instances when the internal modem of a computer had to be removed to get *IC* or *Pcbug* running. The modem used the same interrupt as the robot COM port and disrupted the robot serial communication.

5.2 Serial Communication between Host Computer and TJ PRO™

This section assumes TJ PRO™ has fully charged batteries and the processor functions. Without either of these prerequisites, the user cannot begin.

The MB2325 serial communications board generates the proper voltage conversions that permit the user to download and upload code and data to TJ PRO™ via a 6-wire serial communications link. Whenever testing programs on TJ PRO™, the user will connect the host computer with TJ PRO™ via this interface. Rather than duplicate this interface circuit on each robot, Mekatronix™ chose to make a single external interface for the host computer. This approach has a twofold advantage, 1) removal of this circuitry from the robot platform saves battery energy and reduces printed circuit board cost, 2) only one MB2325 board is required per computer instead of one per robot. In multiple robot systems, or swarm systems, this approach yields an increasing cost advantage.

5.2.1 Plugging MB2325 into the host computer

The MB2325 serial communications board (Figure 3) has a 25 pin, female, type D-connector. On most computers, COM1 is a 25pin male D-connector. If your computer has a 9 pin connector you will need a 9 to 25 pin converter (available at most computer and electronics stores). When the MB2325 is plugged into the COM1 port of the host computer D1 will light. This light lets you know that the serial port on the host computer is working. If D1 does not light, something is wrong with the serial port of the host computer and the robot will not be able to communicate with it..

5.2.2 Connecting the Host computer to TJ PRO™

After the MB2325 is connected to the computer, take the 6-pin rainbow colored cable and insert either end onto J2 of the MB2325.   On the back of the robot is a 6-pin rainbow cable with a male connector. Connect this male header to the female connector on the other end of the cable coming from the MB2325. Now, place the *DOWN-LOAD / RUN* switch to the *DOWN-LOAD* position. Place the *POWER- SAVE / ON* switch to *ON* position. At this point the LED on the robot's top board should light. Press and momentarily hold *RESET* (the red button) on the robot.   With *RESET* pressed, D2 on the MB2325 should light. When *RESET* is released, D2 on the MB2325 turns off.

5.2.3 Verify proper operation

The lights on the MB2325 let you know that the basic connections are working. If on power up, the power light on the top circuit board of the robot did not light, the robot may not be charged. If the batteries are charged, check battery and power connections to the circuit boards. Particularly, check underneath the robot to verify the battery cable has not come undone. If the lights on the MB2325 board do not function as specified, check that no other programs or devices are using the serial port.

5.3 Software Language Support

TJ PRO™ programs can be developed under Image Craft C (ICC11), the compiled version of C for the MC68HC11, or under Interactive C (IC), the C interpreter with multi-tasking developed by Randy Sargent for MIT's 6.270 course under Fred Martin's direction.

ICC11 comes in a DOS and a windows version. The windows version offers an Integrated Development Environment (IDE). Mekatronix™ sells both versions of ICC11 through its distributors.

 IC-DOS is freeware, is supplied at no charge, and can be downloaded from the Mekatronix web site (www.mekatronix.com). IC for WIN95 can be purchased from a Mekatronix distributor. IC for WIN95 allows you to use IC in a windows environment and is much more convenient and easier to use than the IC-DOS version. Either version of IC lacks some application flexibility on the TJ PRO™ as compared to ICC11. For example, in IC you cannot write you own interrupt drivers for add-on motors, servos and sensors.

ICC11 windows version is the preferred programming language for extensive program development on TJ PRO™ as it gives you complete control over all the existing hardware resources and even ones that you add yourself.

IC is useful and easy to use for simple applications. In many cases, the user may prefer to develop robot behaviors on IC. We have attempted to make TJ PRO™ compatible with IC for this reason. We were not wholly successful. IC uses the MC68HC11 "output compare four" (OC4) pin to control its multi-tasking timing. Mekatronix assigns OC4 to control a servo. The rapid changes of OC4 will actually burn out any servo attached to the PA4 header on TJ PRO™'s MTJPRO11 microcontroller board.

WARNING!
IC code thrashes any servo attached to PA4 and will easily destroy the servo in a short amount of time. Disconnected any servo from PA4 before TJ PRO™ runs IC code.

6. INSTALLATION OF ICC11 AND SOFTWARE

If you purchased the ICC11 software this section provides guidance in its installation and integration with the TJ PRO™ software. Be sure to refer to your ICC11 manual for installation particulars for the ICC11 system. The IDE environment makes the *windows* version a sweet tool for programming your Mekatronix robots and makes life much easier than the DOS version. Mekatronix highly recommends the *windows* version. Experience has shown it is well worth the additional cost.

Note: In typed commands to your computer, angle brackets indicate parameters for which you must substitute the appropriate information. Do not actually type the angle brackets. For example, `<enter>` means "Press the Enter key on the keyboard ; `<filename>` means "Type the filename alphanumeric key sequence on the keyboard and the path to it, if necessary."

Summary of Installation Process
1. a. Download ICC11 from the authorized Mekatronix vendor's web site and install, or
   b. Insert ICC11 diskette and install ICC11 and remove diskette when finished.
2. a. Download TJ Pro™ Distribution Software from an authorized Mekatronix vendor's web site and install, or
   b. If applicable, remove ICC11 diskette, insert the most recent version of the TJ PRO™ Distribution Software diskette and install.
3. Installation essentially means copying files into appropriate directories of `c:\icctjp`.

7. INSTALLATION OF ICC11 FOR WINDOWS

Insert ICC11 diskette and install ICC11, or download from an authorized Mekatronix distributor (www.mekatronix.com/distributors) web site. You will see an executable file such as `v51win.exe`. Execute it and follow directions. Specify `c:\icctjp` as your root directory. Let the installation update your DOS CONFIG file. Reboot. Put the ICC11 icon shortcut on your desktop.

After ICC11 installation, you should observe the following directory structure

- `c:\icctjp`
  - `Bin`

- `Examples`
- `Include`
- `Lib`
- `Libsrc`

7.1   IDE Compiler and Linker Setup for TJ PRO™

Once you setup the IDE environment, you will not have to change it from invocation to invocation, since it remembers its most recent state.

1. Double click on ICC11 icon to enter the Integrated Development Environment (IDE).

2. Under *Options* in the menu bar select *Compile*.

3. In *Compiler* window select the Linker tab.

4. In the *Linker* setup window enter the following:

   a. Text section: 0x8000
   b. Stack: 0xffbf
   c. Data Section:
   d. Additional Libraries: libtjp

(You may wish to use the Setup Wizard to save this configuration under the name TJ PRO.)

If you specified `c:\icctjp` as your root during installation, then

   e. Library Path: `c:\icctjp\Lib`

will be the default.

5. Select Preprocessor under *Compiler*:
If you specified `c:\icctjp` as your root during installation, then

        Include Paths: `c:\icctjp\include`

will be the default.

6. Close *Compiler* window.

Continue immediately to the next section.

7.2   IDE Setup for Downloading into a Robot

1. Select *Target* on main menu bar.

2. Select *Bootstrap Download Mode*.

3. Select *Target* on main menu again.

Verify that *Bootstrap Download Mode* has a check mark by it and then

4. Select *Terminal*.

    A window opens. Expand it to fill your screen. At the bottom of the window, you will see several selection buttons. These buttons allow us to configure for bootstrapping loading.

5. Select *Bootstrap Options*. The *HC11* window opens.

6.   In *Bootloader Programming* select "External RAM".

7. Set baud rate to default (1200)

8.   Close *HC11* window.

You are now ready to edit, compile and download programs to your robot from the IDE.

7.3 Integrating TJ PRO™ Software with ICC11 for Windows

Integrating TJ PRO™ software amounts to copying files into `icctjp` directory at the appropriate places. Refer to Section 12.1.

8. COMPILE AND EDIT ON IDE

1. Select *File* in the menu bar and click on *Open*.
    Browse to select the source code file of interest, for example, c:\icctjp\tjpcode\avoidtjp.c, and open it.

A window opens which allows you to edit the program. In the example case, the program needs no editing, but normally you would edit your program at this time. Refer to the ICC manual for details on editor commands.

2. From *Window* on the main IDE menu bar, select *Tile* to tile the Status and Editor windows (ALT-T).

3. Select *Compile* in menu bar and then select *Compile to Executable*.

    The compiler compiles the program in the currently active edit window. If no edit window is open, no compilation is possible.

4. Watch the *Status* window. A successful compile ends with the word SUCCEEDED. Otherwise, note errors, correct your program in the edit window, and repeat Steps 3 and 4.

9. DOWNLOADING WITH THE HSSDL11

The Mekatronix product, the extremely low cost High-Speed-Serial–Downloader-11(HSSDL11), provides an ultra fast (115.2Kbaud) serial download rate from your PC to the robot. This greatly

reduces your development time and improves your productivity. Contact a Mekatronix distributor for purchasing.

Mekatronix encourages the use of the HSSDL11 software for downloading code onto the TJ PRO™. The HSSDL11 operates in WIN95 and can be opened simultaneously with the IDE window. You can conveniently compile and edit in IDE. After downloading using the HSSDL11, you can open up *Terminal* in IDE to receive data from the robot if the program produces serial output.

NOTE: Terminal in IDE and HSSDL11 both use the same COM port (default: COM1). The HSSDL11 releases the COM port after a download, even with the application window still open. Hence, HSSDL11 can remain open when Terminal *in IDE is opened. However, you cannot download from HSSDL11 when* Terminal *or any other device captures the COM port. You will have to close the* Terminal *window or other device holding COM before HSSDL11 can download.*

9.1 Installation
Just copy the HSSDL11 directory with all its subdirectories and files into you hard drive and establish a shortcut to it from your desktop.

9.2 HSSDL11 Setup
The HSSDL11 remembers its setup, so you will only have to perform this procedure once or whenever you want to make changes. The currently selected .s19 file name appears above the HSSDL11 menu bar for convenience. Repeated compiles and loads of the same program during development require only pressing the HSSDL11 menu *Download* button or the d-key on the keyboard. For more HSSDL11 downloading details refer to Sections 9.3.

1. Open the HSSDL11.

2. Confirm default *Settings*:
    Clock: 8MHz
    Port: COM1

3. Click on *Config* button and verify the default settings:

    CONFIG: DO NOT CHECK ANY ITEMS HERE! Default: NO item is checked.

    Features: Check only High Speed option.
    Some older computers will not allow you to communicate at 115Kbaud and you may have to "unclick this option". The baud rate then defaults to 9600.

4. Click on *Config* button again to leave configuration mode.

9.3 HSSDL11 Download Procedure

Be sure the robot power switch is on, the robot power light is on, the physical serial connection is installed, the HSSDL11 has been setup and its window open.

1. If necessary, click on *Program* button to browse and find the TJ PRO™ ".s19" file of interest. Select *Open* to open the desired .s19 file. For example, `c:\icctjp\tjpcode\taltst.s19`.

2. Flip the robot's Download/Run switch to Download.

3. Press the robot's red Reset button.
   Make sure diode D2 on the MB2325 board lights when reset is held down and goes off when reset is released.

4. Select *Download* in the HSSDL menu.
   First, the 256 bytes of the bootloader file loads at 1200 baud. Next, your program loads at 115.2Kbaud into the robot.

You are now ready to execute the code on the robot.

9.4 HSSDL11 Operation Tips

Often the download will fail the very first time you try it. In fact, it may take two or three tries initially. Thereafter, downloads usually succeed on the first attempt.

Precaution: A PC executing background programs may loose synchronization with the robot serial port at these high speeds, so you may need to close such programs if you get repeated boot failures.

We have found that under normal operations with several programs and windows open, the HSSDL11 has no problems.

If the Download prints error message that the COM port is not available, it means that some other application is using COM1 and must be closed before the HSSDL11 can operate.

Sometimes you will forget to press the reset button on the robot or to place the robot in download mode. The HSSDL11 will signal a boot error! Check your switch settings, press reset and verify the D2 light on the MB2325 board flashes on when the reset button is held down. Try again. Typically, after a failure, the HSSDL11 with fail a second time, even with everything correct. Do not worry. Press reset and repeat the *Download* command. For some reason the HSSDL11 sometimes remembers immediate past failures and then recovers on the second attempt, usually. Of course, this problem is avoided altogether if you don't make a mistake with the switches!

10. DOWNLOADING USING IDE

You must carry out the setup procedures in Section 7 before using IDE to download any ".s19" files into the robot. Your setup will be in the state that you left it in at the end of your previous IDE session.

Connect your PC, serial cable, MB2325 communications board and the six-wire Mekatronix serial cable C2325 to the robot as described in Section 14.5.

Open IDE by double clicking on your ICC11 icon.

1.   Select *Target* in the menu bar.
2.   If *Bootstrap Download Mode* is checked, select *Terminal* and skip Step 3. Otherwise select *Bootstrap Download Mode* and do Step 3.

3.   Select *Target* again and click *Terminal*.

You now have the terminal window open.
4.   If necessary, select *Browse* to find the .s19 file of interest. Select the desired .s19 file.

5.   Flip the robot's Download/Run switch to Download.

6.   Press the robot's red Reset button.

     Make sure diode D2 on the MB2325 board lights when reset is held down and goes off when reset is released.

7.   Select *Bootstrap Download* in the IDE *Terminal* window.

8.   Select OK on the IDE message.

9.   The .s19 file loads into the robot.

You are now ready to execute the code on the robot.

11. EXECUTION OF ROBOT CODE ON IDE

The IDE terminal simulator (*Terminal*) accepts robot serial output directly! This allows you to monitor robot IO while the robot remains connected to the PC, a very convenient feature of IDE.

Execute Program Procedure
1.   Download .s19 file into robot from *Terminal* window (refer to Section 9).
2.   Flip robot Download/Run switch to Run.
3.   Press robot red Reset button.

Gainesville, Florida   www.mekatronix.com   Tech Questions: tech@mekatronix.com

CAREFUL PROGRAM WILL START IMMEDIATELY!

Your program will start to run. If your program transmits serial output, it will appear on the terminal simulator of IDE that you currently have open. If your robot produces IO to the *Terminal* screen, you may have to press reset several times or hold it down for a second to get clean communication.

12. TJ PRO™ DISTRIBUTION SOFTWARE

The distribution software includes files with important instructions and information. This information, with some additional comments and elaboration, is repeated here for your convenience. Being familiar with the contents of this package will help you find the information you need to install and use it with your TJ PRO™.

12.1 Installation Instructions for TJ PRO™ Distribution Software

This installation assumes that your C-Compiler has already been installed into the directory `icctjp` and that the batch file installtjp.bat will executes from the directory holding the TJ PRO™ Distribution Software. Execute the installation batch file with an argument specifying the root directory where the TJ PRO™ files will be loaded. If no directory is specified, c:\icctjp is assumed. The latter is recommended as all Mekatronix batch files assume the root is c:\icctjp.

12.1.1 File Name: installtjp.bat

This batch file integrates the TJ PRO™ libraries and code with the compiler installation. See below for a listing of this file.

installtjp.bat

```
@echo off

if "%1"=="" goto noargument

copy tjpsetup.bat %1
copy tjpReadme.txt %1
copy tjpProg.txt %1
copy tjp_Install_Readme.txt %1
copy IDEsetup.txt %1

copy bintjp\*.* %1\bin\*.*
copy incltjp\*.* %1\include\*.*
copy libtjp\*.* %1\lib\*.*

if not exist %1\libsrtjp md %1\libsrtjp
copy libsrtjp\*.* %1\libsrtjp\*.*

if not exist %1\tjps19 md %1\tjps19
copy tjps19\*.* %1\tjps19\*.*

if not exist %1\tjpcode md %1\tjpcode
copy tjpcode\*.* %1\tjpcode\*.*
```

```
md %1\pcbug11
copy pcbug11\*.* %1\pcbug11\*.*

md %1\assembler
copy assembler\*.* %1\assembler\*.*

goto end

: noargument

copy tjpsetup.bat c:\icctjp
copy tjpReadme.txt c:\icctjp
copy tjpProg.txt c:\icctjp
copy tjp_Install_Readme.txt c:\icctjp
copy IDEsetup.txt c:\icctjp

copy bintjp\*.* c:\icctjp\bin\*.*
copy incltjp\*.* c:\icctjp\include\*.*
copy libtjp\*.* c:\icctjp\lib\*.*


if not exist c:\icctjp\libsrtjp md c:\icctjp\libsrtjp
copy libsrtjp\*.* c:\icctjp\libsrtjp\*.*

if not exist c:\icctjp\tjps19 md c:\icctjp\tjps19
copy tjps19\*.* c:\icctjp\tjps19\*.*

if not exist c:\icctjp\tjpcode md c:\icctjp\tjpcode
copy tjpcode\*.* c:\icctjp\tjpcode\*.*

md c:\icctjp\pcbug11
copy pcbug11\*.* c:\icctjp\pcbug11\*.*

md c:\icctjp\assembler
copy assembler\*.* c:\icctjp\assembler\*.*

: end
```

12.2 File Name: tjpReadme.txt

This file summarizes the contents of the distribution software package.

```
**************************************************************************
BE SURE TO REFER TO THE TJ PRO USERS MANUAL FOR DETAILED INSTRUCTIONS!
**************************************************************************

                  CONTENTS OF THIS DISK

TALKRIK JUNIOR PROFESSIONAL (TM) SUPPORT SOFTWARE
========================================================================
This disk includes the following directories:

<Assembler>       Freeware assembler for the M68HC11
<Bintjp>    Batch files to compile and load TJ PRO programs
<Incltjp>   TJ PRO include files used specifically by TJ PRO programs
<Libtjp>    TJ PRO library files
<Libsrtjp>  Object code for TJ PRO library files
```

```
<PCBUG11>    PCBug11 monitor for the M68HC11, Motorola Freeware
<tjpcode>          Applications Source code for TJ PRO


-------------------------------------------------------------------------
The distribution also contains the following files:

IDEsetup.txt       How to use IDE to edit, compile, download to TJ PRO.
installtjp.bat     Batch file to install the TJ PRO distribution software
           into your ICCTJP directory. Useful for both DOS and
           WINDOWS versions.

tjp_install_readme.txt
           Brief directions on installation of TJ PRO distribution
           software.
tjpProg.txt How to compile and download code into the robot using
           ICC11 DOS.
tjpReadme.txt      This file.
tjpsetup.bat       Batch file to setup paths and ICC11 linker parameters.
-------------------------------------------------------------------------


Assembler directory

This directory contains as11.exe, a freeware assembler for the M68HC11


-------------------------------------------------------------------------
Bintjp directory

This directory contains the batch files comptjp.bat, loadtjp.bat, cltjp.bat
which
let you, from a DOS window, to
1. Compile a TJ PRO program,
2. Download a TJ PRO program in Motorola ".s19" file format into the robot
using
   PCBUG11 (PCBUG11 has a number of severe limitations. For trouble free
operation,
   ease of use, convenience and more than 10 times faster downloading we
recommend
   you purchase the inexpensive Mekatronix HSSDL11. Our programmers use
nothing else!),
3. Do both a compile and load of a TJ PRO program into the robot,
respectively.


To use loadtjp.bat, your program's ".s19" file must reside in the directory
tjps19
which must be in the root icctjp at the same level as the pcbug11 directory.
-------------------------------------------------------------------------
Include directory Incltjp

This directory contains all the header files <file.h> required by TJ PRO
applications written in C. As you add new applications you will add new header
files to this directory. For example, you can purchase and add the SERVO
software
kit which permits calibrating and controlling up to three servos on the TJ
PRO.


-------------------------------------------------------------------------
```

Library directory Libtjp

This library directory contains the basic TJ PRO library file "libtjp.a"

Library Source directory Libsrtjp

TJ PRO library object files.

PCBUG11 directory

This is Motorola's monitor program for the HC11. Mrc11.bat is used by
loadtjp.bat

Also included is the batch file MSCC11.bat, which will automatically load a
program
into a M68HC11E2 chip.

HIGHLY RECOMMENDED:
   Buy the HSSDL11 (well worth the price!) and do not use PCBUG11.
   It hangs up the COM port and you have to kill the DOS window using
   PCBUG11 in order to use any other serial device on your computer and vice-
versa.

TJ PRO Code directory tjpcode

This MEKATRONIX(TM) directory contains a very simple collision avoidance
program "avoidtjp.c" and the important "sensetjp.c" program. "sensetjp.c"
outputs the sensor readings to your PC each time you type a character. It is
an excellent program for you to familiarize yourself with the robot sensory
capability and to verify sensory operation. Source code is provided in C.
The S19 files "avoidtjp.s19" and "sensetjp.s19" can be downloaded and executed
directly, even if you do not have a C compiler.

----
tjps19

PCBUG11 downloads s19 files from this directory when you invoke the batch file
loadtjp.bat. Path and name length restrictions make it almost impossible to
use
pcbug11 in complex directory structures. The directory name tjpcode is too
long(!)
and cannot be used! Be sure your ".s19" file names are no more than eight
characters long (not counting the extension ".s19") and that this file is in
tjps19
directory at the same level as the pcbug11 directory in the icctjp directory.
Correct
installation will guarantee this. However, creative programmers who prefer a
different
directory structure are advised of this severe path-length limitation aspect
of pcbug11.

12.3 File Name: tjProg.txt

This file provides instruction on how to edit, compile, download, execute your TJ PRO™ robot using ICC11 DOS. If you have purchased ICC11 for Windows, skip this file and read IDEsetup.

IMPORTANT NOTE
If you execute a TJ PRO program that uses functions in serialtjp.c and you loaded the ".s19" file with PCBUG11, then you must close the DOS window in WIN95, because PCBUG11 locks up the COM port. Hyper Terminal will not be able to communicate through the COM port with the robot. For some archaic reason PCBUG11 in DOS hogs the COM port and does not release it unless you kill the DOS window.

SOLUTION
Eliminate PCBUG11 use. Buy the (inexpensive and well worth it!!!) Mekatronix high-speed serial downloader (HSSDL11) for the MC68HC11 and avoid the pain of closing DOS and opening and setting up the ICC11 parameters every time you reopen the DOS window!

When using the HSSDL11 with Hyper Terminal on the same COM port, say COM1, you must disconnect Hyper Terminal (ALT-C d or select the Hyper Terminal telephone icon with receiver off the hook) to download. After download, connect Hyper Terminal (ALT-C c or select Hyper Terminal telephone icon with receiver on   the hook). You do not have to do anything with HSSDL11 since it does not hog COM1, an extremely useful convenience because it retains its state   and lets you quickly download a program you repeatedly compile and change   in DOS. Just switch back and forth between the DOS window and the HSSDL11 window...for convenience keep both open and side by side!

12.4 Editing TJ PRO™ Programs

Use your favorite program editor, but be sure to save the program as a text file with the ".c" suffix.

After editing, you save the program, go to a DOS window, if your computer supports WINDOWS, or a DOS prompt and compile the program. Download your program into the robot and execute. Repeat.

The steps to carry out this process are described in the following sections.

Suggestion
In a WINDOWS environment, open your favorite editor with your program, the HSSDL11, Hyper Terminal and a DOS window simultaneously...this makes a very convenient working environment similar to IDE, but not quite as convenient.

12.5 Compile with ICC11 DOS

If you purchased the ICC11 C DOS compiler, you can write TJ PRO programs in C and compile.

1. Open a DOS window in a WINDOWS environment or boot up in DOS.
2. At the DOS prompt, change the directory to `icctjp`.
3. Execute `tjpsetup`.
4. Type

```
comptjp <filename> <enter>
```

where `<filename>` is a C program in the directory `tjpcode` (default). Do not put the .c extension on the name. The batch file will do that for you. You can also provide a complete path to `<filename>` if it is not in the `tjpcode` directory.

5. To compile other files during a session, repeat only Step 4.

If you are in an exclusively DOS environment, Step 1 is automatic, otherwise the procedure is the same.

12.6 PCBUG11 Download Procedure for TJ PRO™

1. Connect C2325, 6-wire serial communications cable, to the Mekatronix MB2325 serial communications board at one end and to the J54 male header on the MSCC11 at the other end. The MB2325 is connected via a 25-pin D-connector and cable to your Personal Computer. This setup establishes a link from TJ PRO(TM) to your PC. Check to make sure diode D1 lights.

2. Switch DOWNLOAD/RUN Switch to DOWNLOAD.

3. Turn on TJ PRO power with POWER-ON Switch.

4. Press RESET switch. D2 lights when you hold RESET down. If it does not light, reverse the C2325 6-pin connector on J54 of the MSCC11.

5. Be sure the ".s19" file you wish to download is in the tjps19 directory.

6. Go to DOS window and prompt.

7. Perform the Compiler setup sequence (Section 12.5) if this is first time in this DOS window.

8. Type

```
loadtjp <filename> <COM_PORT_NUMBER><enter>
```

For example, type

```
loadtjp avoidtjp 1 <enter>
```

and PCBUG11 will download avoidtjp.s19 into TJ PRO's RAM. The serial communications will be via COM1 (COM1 is the default if no com-port is specified). The cable from the MB2325 board should be connected to COM1 on your PC in this situation.

9. If your program does not produce serial input or read serial input, steps 2. to 8a. can be repeated unchanged after each compile. However, if your program does use serial IO, then you must terminate the DOS window before enabling Hyper Terminal and executing the robot program. When finished with your experiment, you must repeat the four-step sequence COMPILE WITH ICC11 described in the previous paragraph in order to be able to compile again. You cannot simply execute step 4 in that sequence. This is a real inconvenience, so..

12.7 Execute a TJ PRO™ Program

This procedure assumes you have downloaded an ".s19" file.

Do not do this procedure on a Table, unless TJ PRO's wheels are suspended and not touching any surface. Otherwise, TJ PRO may start up and roll off!

1. Turn TJ PRO power on.

2. Flip DOWNLOAD/RUN switch to RUN.

3. Press RESET.

4. Watch TJ PRO GO! (Some programs require pressing the back bumper as a standard precaution.)

NOTE:
If your program uses serial IO and you downloaded with pcbug11, you must close the DOS window before Hyper Terminal can grab COM1 to communicate with the robot.

12.8   Stopping TJ PRO

To stop TJ PRO, run him down and flip the POWER-ON switch off! If your program requires the back bumper to be pressed before it executes, you can also press the red reset button to stop the robot.

12.9 Brief Descriptions of TJ PRO™ programs in directory tjpcode

The ".s19" files for the programs described below can be loaded directly into your robot and executed. You can find further descriptions of the files in the ".c" versions. To really understand the details of each program, however, read the ".c" code itself!

```
avoidtjp.c
```

1. Back bumper must be pressed to start robot.
2. Performs obstacle avoidance.
3. If forward bumpers are pressed during obstacle avoidance
   mode, Talrik backs up, turns a random amount of
   time and then goes forward again.

tjprotst.c

This program allows you to test all of TJ PRO™'s basic features:
IR Detectors
Motors

Select the appropriate test from the startup menu. When finished, press reset on the TJ PRO™
and select the next test.

sensetjp.c

This module reads all the sensors and updates the appropriate variables and displays them on the
screen in a pattern similar to their geometric position on the TJ PRO™ plate.

12.10 TJ PRO™ Library Code in directory Libsrtjp

Before using any of the TJ PRO™ library functions the appropriate initialization programs
must be executed. The init_<program> for the files below are:

```
  init_analog();
  init_clocktjp();
  init_motortjp();
/* No init routine in muxsentjp.c */
  init_serial();
  init_servos();
/* No init routine in Vectors.c */
```

The TJ PRO™ library functions contain these programs. A brief description of the TJ PRO™
library functions appears below.

analog.c

  Defines init_analog() and analog() functions. analog() reads the analog
  to digital converter output on ADR1.

clocktjp.c

  Generates msec, seconds, minutes, hours, days. Also, variable timertjp is a
  msec free running counter which can be used. The function wait() permits
  your programs to execute time waits.

motortjp.c

Drivers for TJ PRO™ motors.

serialtjp.c

Defines the Mekatronix serial functions. Use `<stdio.h>` whenever possible, but
the function `read_int()` here is quite useful. Also, executing `init_serial()` will
initialize appropriately for <stdio.h> as well.

servotjp.c
     (Option. Not included in TJ PRO™ Distribution Software.)

Driver for servos attached to PA4, PA5, and PA6 on the TJ PRO™ MTJPRO11 microcontroller
circuit board.

Warning!
Do not attach anything to PA4 when running Interactive C.

Vectors.c

Defines all the MC68HC11 interrupt vectors generically. User and future Mekatronix interrupt
service routine names must equate to the predefined generic names.

12.11   tjpsetup.bat file and the TJ PRO™ Library Libtjp.a

The `tjpsetup.bat` file defines the paths to the `Include` and `Lib` directories as well as establish
the necessary linker options. Observe that the code begins at 0x8000 and the stack at 0xffbf. Note
that the MC68HC11 stack grows toward lower memory addresses, hence, the stack grows
toward your data and code sections and could potentially cause havoc with your program if it gets
too large. Be aware of this possibility for large programs.

Warning!
Always be ready to turn the robot off immediately if it seems to be oscillating rapidly and out of control or stuck.

The library option `-ltjp` in the linker setup is essential and refers to `Libtjp.a`. The library
facility of ICC11 compiler versions 5.0 and greater is not compatible with the same facility of
versions less than 5.0. A safe rule is that an ICC11 compiler will compile TJ PRO™ code if
`Libtjp.a` has been compiled on that same version. In any case, you can also add object files to
this library following the template in `Libtjpmake.bat`.

Gainesville, Florida   www.mekatronix.com   Tech Questions: tech@mekatronix.com

 If you have the DOS version of the compiler and the library source code[3],you can remake the library `Libtjp.a.` with `Libtjpmake.bat`.

Contents of `talsetup.bat`:
```
doskey
PATH %PATH%;C:\ICCTJP\bin
set ICC_INCLUDE=C:\ICCTJP\include
set ICC_LIB=C:\ICCTJP\lib
set ICC11_LINKER_OPTS=-btext:0x8000 -ltjp -dinit_sp:0x7fff -dheap_size:0
```

12.12 Generation of TJ PRO™'s Library Libtjp.a with DOS Script Libtjpmake.bat

The distribution code shipped to you will provide `libtjp.a` compiled in a compatible version of the compiler you purchased and you do not need to do anything here. This section is for experts who wish to add object files or remake `libtjp.a`. You must have source files or object files to remake or add files to a library. If you have source files, just compile to object files and add. Refer to the ICC11 manual for details.

For the DOS version use the batch file **`Libtjpmake.bat`** (listed below). For the Windows IDE version, you simply duplicate the process indicated by the batch file.

Contents of `Libtjpmake.bat`:
```
echo off
echo This script file makes a library file for the TJ PRO Routines
echo Title        Make TJ Pro Libraries
echo Programmer      Keith L. Doty
echo Date         Sept. 15, 1998

icc11 -c vectors.c
icc11 -c serialtp.c
icc11 -c motortjp.c
icc11 -c analog.c
icc11 -c clocktjp.c
del ..\lib\libtjp.a
ilib -a libtjp.a  serialtp.o
ilib -a libtjp.a  motortjp.o
ilib -a libtjp.a  analog.o
ilib -a libtjp.a  vectors.o
ilib -a libtjp.a  clocktjp.o
ilib -t libtjp.a
copy libtjp.a ..\lib\*.*
```

13. PROGRAMMING YOUR TJ PRO™ WITH INTERACTIVE C

Whether you have the commercial or freeware version of IC, usage is similar, although the commercial version is much more convenient and has more features. The following sections explain how to program TJ PRO™ using Interactive C.

---

[3] The TJ PRO™ library source code is not included in the Distribution Software and comes as a separate option in the TJ PRO™ Education software package for ICC11. Call your Mekatronix distributor for details.

13.1 Installing and Using IC

You can download the freeware IC DOS and the IC manual from the Mekatronix web site under *Free Stuff* without charge.

Before installing IC, first check a few things.

Check List.
1. Does your robot have batteries in it and are they charged? If this is the first time you are using your batteries, they should have been charged overnight. When you turn the power switch on, the small red light should come. Even with low batteries this light may come on, so, if you have not charged your robot, please do so now or install fresh batteries.

2. Does your computer run DOS or WINDOWS? Does your computer have a serial port available? There should not be conflicting hardware demands on the serial interrupt. For example, if you are using COM1, COM3 should be left free. Likewise with COM2 and COM4. We will assume here that you are using COM1.

3. Do you have a serial cable coming from your serial port that ends in a 25 pin D connector? You will need this cable to connect to your MB2325 board as shown in Figure 1.

4. Install IC

   IC WIN95.
   　　Follow installation instructions. Name your root file `c:\ictjp`.

   IC DOS.
   　　Create a directory on your computer for IC on DOS:
   　　At the DOS prompt　　　　**type**　　　　*mkdir ic*
   　　Change to the IC directory,　**type** :　　*cd ic*
   　　Download IC DOS from Mekatronix web site into `c:\ic` directory.

5. Download the *TJ PRO™ Distribution Software* from the Distributor's web site as instructed by the Distributor. Copy the distribution files into the `ictjp` or `ic` directory, respectively, for WIN95 or DOS versions.

After completing steps four and five, you are ready to start talking to your robot.

13.2 Serial Communications with the TJ PRO™.

To serially communicate with your TJ PRO™ follow the following steps:

1. Place your TJ PRO™ next to your computer and support it so the wheels don't touch the table (a roll of electrical tape works great for this). Jacking up the wheels prevents the robot from driving off the table when the motors turn on!!!

2. Connect the MB2325 to your serial cable the light labeled D1 should light. If this works proceed to step 3.

3. Connect the C2325 cable to the MB2325 on the connector labeled J2 with the blue wire toward the J2 label. Now connect the other end of this cable to your robot, through the hole under the word *Mekatronix* with the blue wire to the right. Be careful not to offset the cable, all six pins should connect to the six wires of the cable.

13.3 Loading Pcoder with IC WINDOWS

IC was designed originally to execute on the MIT Handy Board, which has a rather obscure hack on the serial input. IC takes advantage of that hack, but it makes for strange initial loading of the system on aTJ PRO.

1. Turn on the robot and place the *DOWNLOAD/RUN* toggle switch into *DOWNLOAD*.

2.   From Windows execute IC.

    IC will give an error message that the board is not responding. Not to worry! The board is not responding because you have not had a chance to load the *p-coder* and relevant libraries into the robot.

3. Select the *Yes* button in the IC error Window to configure the board.

    The system is configured to enable COM1 for serial communication. If you wish to use another COM port select it on the screen that now appears. To make your selection the default, you will need to change the `Port = 1` command in the `ic.ini` file to `Port = n,` where n is the COM port number you picked.

4. Select the *Download Pcode* button in the opened IC Window.

    IC will now open the `Libs` directory.

5. Select the file `TJ-pro.icd` file to open.

6. Press the red *RESET* button on the robot and select the *OK* button in the window.

    You will observe a flurry of loading activity with the green LEDs on the MB2325 board rapidly flashing, and then another error message☺! IC cannot communicate with the p-coder program because the robot is in *DOWNLOAD* mode and not *RUN* mode!

7. Place the robot *DOWNLOAD/RUN* toggle switch into *RUN*.

8. Press the red *RESET* button.

9. Select the *Yes* button in the IC error Window.

    IC will then load the remaining code onto the robot and you are ready to go.

As long as you keep power to the TJ PRO™ memory, the *p-coder* will not have to be reloaded and you can turn the power switch on-and-off at will. If the batteries drain or you otherwise overwrite memory you will have to repeat the *pcoder load procedure* described above.

13.4 Loading Pcoder with IC DOS and other Methods

You can load the IC virtual machine pcoder software `Pcode_rw.s19` using any of the three downloading methods discussed earlier:

1. HSSDL11 (Section 9) (Recommended)
2. ICC11 IDE (Section 10)
3. PCBUG11 (Section 12.6)

Or you can use the `init_tj` method described here.

If you use any of the above three methods be sure load the files
```
twoservo.icb
twoservo.c
motorp.c
```
before you load any program.

1. To load the IC DOS pcoder into the robot with `init_tj` method you must boot up in DOS, change the directory to `c:\ic` and then

If you are using COM1	**type:**	`init_tj 1`
If you are using COM2	**type:**	`init_tj 2`

    This action will download the brains of IC, the pcode interpeter known as the *pcoder*. During the transfer you should see the D1 and D2 lights on the MB2325 board blink rapidly. After 30 seconds the screen should prompt you to "press any key to continue" at this point switch the Download / Run switch to Run and press reset. Then press any key on the keyboard.

2. Place the Download / Run switch to the Download position and turn the robot on with the power switch. The red light should come on. Press the reset switch. When you press the reset switch, the light D2 on the MB2325 should come on. When you release the switch it should go off. If this happens continue, otherwise check connections, power, and cable orientation.

3. After a few seconds you will see `default.lis` and other files loading. Next, you will see a *C>* prompt, indicating that your robot is now running IC. On the other hand, if you see

synchronizing with board messages, then press Ctrl-C on the keyboard and repeat Step 2, or you may have to start all over.

13.5 Loading and Unloading Programs in IC

Once the pcoder has been loaded you can reboot your system and work from a DOS window when running IC. This allows you to open an edit window without exiting IC as you would in DOS only mode. Of course, if DOS is all you have, that's what you will do!

1. Now that you have loaded IC, you can load a test program. At the IC prompt, this is the interactive *Command Line* of IC,

    type: load tjpro1.c

2. When you get the *C>* prompt back, you can run the robot on a test stand you have devised. Or, you can disconnect the serial cable and the power jack and place your TJ PRO™ on the floor. To run the program press the reset button and watch TJ PRO™ go! (Unless, of course, you have programmed the robot to only begin after the back bumper switch is closed.)

To get back to IC and start another programming session,

3.   Turn the power off. Pick the robot up and plug it back into the C2325 cable and plug the AC adapter power jack into the robot power connector.

If you do not use a back bumper start command, put the Download/Run mode switch into Download, turn on the robot power with wheels suspended off the table and reset. The robot will stop executing the current program. You must remember to switch the robot back into Run mode and press reset before resumption of IC commands. If you use the back bumper as a Start command in your robot programs, you do not have turn power off to stop the robot, just press reset. Since the robot is in Run mode, the IC command line remains active. This is quite handy and Mekatronix recommends you program Start this way. Most future Mekatronix code will be written with a Start command.

4. In a DOS environment you will have to exit IC to edit your program. When finished editing, exit your editor, return to IC by executing `ic` and then type the IC command `load <file>.c` into the IC command line. If the file is different from the previous file you must first unload the previous file with the command `unload <file>.c` and then load the new file with `load <new_file>.c`. In the example, to load `sensetjp.c` you first must unload `tjpro1.c`:

    type: unload tjpro1.c
    type: load sensetjp.c

5. In a WINDOWS environment, you will not have to exit IC, simply switch back and forth between the DOS window executing IC and your editor window.

13.6 Loading and Unloading Programs in IC WINDOWS

The procedure for loading and unloading code in IC WINDOWS is essentially the same as for IC DOS, with several major differences. In IC WIN95 you never leave the windows environment. You do not have to boot up in DOS to load the pcoder and then back to WINDOWS to open both a DOS window and an edit window. The browser on IC WIN95 also gives you complete freedom to load `<file>.c` located in any directory. IC WIN95 implements more C programming functionality and is supported by Mekatronix.

13.7 Interactive C

Now that you have learned to run IC on your robot and load programs to the robot, let us develop a greater understanding of the robot and how to write software for it. Interactive C is just that, interactive. At the *C>* prompt you can type any IC command in standard ANSI C format. Each line will return a value. For example, type the command:

> *C>* 1+2;

IC returns

> <int> 3

This value was calculated on the robot's own computer *not* on your PC.

13.8 Robot Commands

There are several commands in IC that are not present in ANSI C, These commands have important uses in robots. Here we discuss the motor driver commands. Since the wheel drive motors on the TJ PRO™ are actually modified servos, their actuation is controlled by IC's servo *command, and not IC's* motor *command. Since this nomenclature and usage may confuse the programmer, Mekatronix provides a routine called* motorp.c *to control TJ PRO™'s motors. The file* motorp.c *is automatically loaded into the* default.lis *file when IC is loaded.*

Before turning on the motors make sure the robot's wheels are held off the table (or else your TJ PRO™ will dive off the table). The motors are enabled with the command

```
servo_on();
```

The servos will probably spin as soon as you turn them on with the above function, so it is wise to follow with motorp(0,0) *and* motorp(1,0) *to definitely stop the wheels while the robot is at rest:*

```
servo_on();
motorp(0,0)
motorp(1,0)
```

 A*fter turning the servos on, you can use the commands* motorp at will*:*

`motorp(1,100);`	turn on the right motor to full forward, ie 100 percent.
`motorp(1,-100);`	turn on the right motor to full backward, ie -100 percent.

     `motorp(0,50);`   turn on the left motor to 50% forward speed.

Note that a negative number in the speed will cause that motor to go backwards.

IMPORTANT!
ULESS YOU CHANGE THE MOTOR SPEEDS WITH A DIFFERENT motorp() COMMAND, THE MOTORS WILL RUN AT THE LAST SPECIFIED SPEED FOREVER.

WARNING!
SPEED PERCENTAGES ARE NOT LINEAR! FOR EXAMPLE, 50% WOULD BE HALF SPEED IN A LINEAR SCALE, BUT IT IS NOT. YOU CAN CALIBRATE THE MOTOR SPEED CONTROL OF YOUR ROBOT IN SOFTWARE. ALTHOUGH YOU CAN IMPROVE THE SPEED CONTROL CONSIDERABLY THIS WAY, BE WARNED THAT EVEN WITH CALIBRATION THE MOTORS SPEEDS CANNOT BE PRECISELY CONTROLLED.

The 68HC11 has eight analog input ports. The command analog(port_number) reads these ports. For example, to read the left analog IR:

     type: `analog(3);`

This will normally read in the high 80s to low 90s with the IR emitters off. If you illuminate the area around the robot with 40kHz IR light, the values will increase to a maximum (125-130) when a white surface is placed close to the robot. The IR LEDs are connected to a memory mapped port on the robot. The poke command may be used to turn them on:

     type*:*   `poke(0x7000,0xff);`

The first number in the poke command is the hex address of the port and the second number is the hex byte to be written to that port in this case it is all high therefore all the LEDs will turn on.

13.9 *Writing your own IC Programs*

Writing programs in IC is very similar to *Standard C*, with a few exceptions. For example, there is no need to use the *include* statement, since all required functions must be loaded with your program. Neither the *Standard C* input-output functions nor IC's *printf* function for an LCD screen display, make sense on your TJ PRO™. Mekatronix resolves this problem with the special code file `serial.c` to allow the TJ PRO™ to send data on the RS-232 bus to your PC. Since IC also uses the serial port, a program that uses the serial port will usually cause communication malfunction. The solution is to exit IC while the robot runs a program that requires IO. To enter program mode again you will have to restart IC as outlined previously.

When the power switch is turned off while TJ PRO™ is plugged into the charger, power is still supplied to the SRAM to prevent loss of code and data in memory. This feature has a decided

advantage. Programs in storage will be there at power on. If the *pcoder* was loaded in your last programming session you can conveniently restore it by simply typing IC.

13.10 Programming Suggestions

Mekatronix™ provides an elementary program that allows TJ PRO™ to explore his environment and avoid bumping into things, most of the time! If TJ PRO™ does bump into something, his bumpers tell him and he moves away.

You can develop your own applications, limited only by your imagination!

1. Make TJ PRO™ do a figure eight, or any other shape, while at the same time avoiding people and furniture.
2. Program TJ PRO™ to be an artist who draws on cardboard with a pen attached to his body (penholder not included) (Be sure TJ PRO™ stays on the cardboard!).
3. Design an obstacle course for TJ PRO™ to learn.
4. Scare TJ PRO™ by blasting him with your TV remote!
5. Write a program so TJ PRO™ will be attracted to your TV remote!
6. Control TJ PRO™'s behavior with your TV remote…an IR controlled vehicle! (A Mekatronix kit sold separately [See http://www.mekatronix.com]
7. Get two or three TJ PRO™'s and program them to follow each other.
8. Get three TJ PRO™'s and teach them to flock like goslings as they move around together.

13.11 Mekatronix IC Education Package

To learn more about how to use IC and to program your TJ PRO™ robot in IC, you may wish to obtain Mekatronix's IC education package. The package includes a 57-page manual with program exercises and source code solutions that load and execute. Contact an authorized Mekatronix distributor for more information.

14. PROGRAMMING YOU TJ PRO™ IN ICC11

ICC11 is a commercially supported C compiler for the MC68HC11 processor. ICC11 with the required robot library files is only available from an authorized Mekatronix distributor. The following paragraphs indicate how to incorporate ICC11 into your TJ PRO™ software development activities.

14.1 Installation of ICC11 and TJ PRO™ Software

If you purchased the ICC11 software from Mekatronix, this section provides guidance in its installation and integration with the TJ PRO™ software. Be sure to refer to your ICC11 manual for installation particulars for the ICC11 system.

Note: In typed commands to your computer, angle brackets indicate parameters for which you must substituted the appropriate information. Do not actually type the angle brackets. For

example, `<enter>` means "Press the Enter key on the keyboard" ; `<filename>` means "Type the filename alphanumeric key sequence on the keyboard and the path to it, if necessary."

1. Download ICC11 from authorized Mekatronix vendor's web site or, if applicable, insert ICC11 diskette into your floppy drive. Execute the startup and follow directions as outlined in the ICC11 manual. Install ICC11 in your root directory as suggested. This eliminates some subtle directory addressing problems with PCBUG11. When you finish, you should observe the following file directory structure

   - `C:\ICC11`
     - `Bin`
     - `Examples`
     - `Extra`
     - `Include`
     - `Lib`
     - `Libmtask`
     - `Libsrc`

14.2 Integrate TJ PRO™ Software with ICC11

1. Download the TJ PRO™_Distribution_Software from your Authorized Mekatronix vendor's web site or, if applicable, insert a TJ PRO™_Distribution_Software Diskette for ICC11 into your floppy drive. In DOS or Windows95 DOS execute

   ```
   install.bat <path to ICC11 file><enter>
   ```

   If no parameter is entered for `<path to ICC11 file>` the default is `C:\ICC11`. This batch file integrates the TJ PRO™ software with the ICC11 software.

   After installation of the TJ PRO™ software, you will see three new directories under ICC11:
     - Assembler
     - pcbug11
     - tjpcode

   The <include> and <lib> directories will also contain the TJ PRO™ include and library files.

IMPORTANT!
ICC11 will not run on the TJ PRO™ robot without the TJ PRO™ distribution software for ICC11.

2. Always execute the modified `iccsetup.bat` when opening a DOS window. Do not execute it more than once in an opened DOS window.

After installing the TJ PRO™ Distribution Software and ICC11, the user can write C programs, compile them and download their S19 file output to TJ PRO™ to test, debug, and run. A convenient method for initial testing of programs on TJ PRO™ is to place TJ PRO™ next to the host computer on an elevated platform with the drive wheels suspended in the air so they do not contact any physical surface. After the user develops confidence in the program's operation, the user typically makes a closely monitored trial run on the floor. Remember, TJ PRO™ is autonomous and the behavior you expect may not happen!

14.3 Compile a TJ PRO™ Program with ICC11

1. On Windows95 open an MSDOS window and execute `iccsetup.bat`. Assuming a DOS prompt, set the current directory by typing

```
cd <path to icc11>\icc11\tjpcode<enter>
```

Fill the path parameter `<path to icc11>` with the path to `icc11`. If you placed the ICC11 files in the root, then the default path is `C:\`. The `tjpcode` directory is where we suggest you put your TJ PRO™ programs and libraries.

2. Type: `comptjp <path\filename.c> <enter>`

   to compile a C source file.

   NOTE: DO NOT PUT THE .c EXTENSION ON THE FILE NAME.

   The path to `<filename.c>` must be the current directory or be explicitly specified or established previously by `PATH`.

14.4 Example Compilation of a TJ PRO™ JUNIOR PROFESSIONAL C Program

 Assume the current directory is `tjpcode` and that `iccsetup.bat` was executed once when the DOS window was opened.
   Type:
```
compTJPRO Avoidtjp.c <enter>
```

   to compile the sample program `TJPROavoid.c` in `TJPROcode` directory.

14.5 Download a Compiled/Assembled Program into TJ PRO™

There are two methods for downloading ".s19" files into the TJ PRO™. You can use Mekatronix's High-Speed-Serial-Downloader 11 (HSSDL11) or PCBUG11. The super fast HSSDL11 downloads programs at 115.2Kbaud and is the preferred method of downloading. The HSSDL11 may be purchased from an authorized Mekatronix distributor.

Before downloading connect the serial communications cable between your robot and personal computer as explained next.

14.5.1 Serial Connection between TJ PRO™ and you Personal Computer

1. Connect one end of a C2325, a 6-wire serial communications cable, to the Mekatronix MB2325 serial communications board. There is only one 6-pin male header on the MB2325, just below the two LEDs. Connect the other end of this cable to the serial port on the MSCC11 through the oblong slot on the TJ PRO™ plate underneath the lettering Mekatronix (Figure 5). The header seen in this serial port slot is the J54 male header on the MSCC11. In Figure 5, corresponding pins of the 6-pin headers on the MB2325 and the TJ PRO™ plate, match from left-to-right.

Figure 5 This Photograph shows the gray serial cable from a PC COM port mating with the D-25 connector on the communications board (com-board = MB2325 = the exposed circuit board sitting on the white boxes). The multicolored 6-wire serial cable attaches to the male header on the com-board and into the serial slot on the TJ PRO™ plate. Note the same color orientation of both ends of the 6-wire cable for the configuration pictured.

3. The MB2325's 25-pin female D-connector can connect directly to your Personal Computer COM port or via a serial cable. Make such a connection. This setup establishes a link from TJ PRO™ to your PC. Check to make sure diode D1 lights when you connect the MB2325 to your PC via the serial cable or directly into its serial port. The advantage of using a serial cable between the MB2325 and COM port now becomes obvious, you can easily see the LED status lights on the MB2325. Refer to Figure 2 for switch and connector locations.

4. Switch DOWNLOAD/RUN Switch to DOWNLOAD.

5. Turn on TJ PRO™ power with the ON/OFF switch flipped to ON.

6. Press the red push-button RESET switch. D2 lights when you hold RESET down. If it does not light, reverse the C2325 6-pin connector to the MB2325.

With the serial connection made you can proceed with downloading programs into the robot.

14.5.2 Downloading with the HSSDL11

1. In a WINDOWS environment activate the HSSDL11 by double clicking the icon. You may want to put a shortcut on your desktop.

2. In the HSSDL11 press or select PROGRAM, browse until you find `tjpcode`, or whatever directory contains the ".s19" file of interest. Open the ".s19" file of the program you wish to download, say, the one you just compiled. There are no name-length restrictions for the HSSDL11, but there is for PCBUG11!

3. Press the *d* key or select DOWNLOAD button and the green lights flash and the bootloader is loaded at 2100baud and then the program downloads at a blistering 115.2KBAUD!

14.5.3 Downloading with PCBUG11

NOTE: PCBUG11 usually locks up the COM port on your computer, so you will have to close the MSDOS window before regaining control of the port. In some instances you may have to restart you PC to unlock the COM port!

1. In WIN95 open an MSDOS window. If you are using ICC11, then execute `iccsetup.bat`. Assuming a DOS prompt, set the current directory by typing

```
cd <path to pcbug11>\pcbug11<enter>
```

Instead of typing `<path to pcbug11>`, you may prefer to include the path to PCBUG11 in PATH.

2. In the following command `<sp>` means type a space. Be sure the Download/Run switch is in Download position. Type

```
pcbug11<sp> -a <COM_PORT_NUMBER><enter>
```

If the robot is connected to COM1, then the command is

```
pcbug11 -a 1
```
This command puts the DOS window into the PCBUG11 Monitor window with PCBUG11 command line at the bottom of your screen with command prompt `>>`.

3. Step 1 in this sequence placed your computer in the `PCBUG11` directory. If the TJ PRO™ serial cable is connected to COM1, then execution of the next PCBUG11 command will download the file avoidtjp.s19 in the tjpcode directory via COM1 to the TJ PRO™ memory. In PCBUG11 execute

```
>>loads icctjp\tjpcode\avoidtjp 1<enter>
```

Note: Your path to the code may be different than the above illustration.

Gainesville, Florida   www.mekatronix.com   Tech Questions: tech@mekatronix.com

14.6 Execute a Loaded TJ PRO™ program

This procedure assumes you have downloaded an ".s19" file into TJ PRO™.

Do not do this procedure on a Table, unless TJ PRO™ 's wheels are suspended and not touching any surface. Otherwise, TJ PRO™ will start up and roll off!

1. Turn TJ PRO™ power switch to ON, if it is not there already.

2. Flip DOWNLOAD/RUN Switch to RUN

3. Press RESET.

4. Watch TJ PRO™ GO!

To stop TJ PRO™ , run him down and flip the ON/OFF switch to OFF.

15. TALRIK JUNIOR PROFESSIONAL™ SOFTWARE REFERENCE

This section is designed as a quick reference to your TJ PRO™ robot distribution software that integrates with ICC11. This section assumes that you have a working knowledge of ANSI C, and will guide you through the drivers and subroutines provided by Mekatronix, which allow you to write TJ PRO™ software ranging from the simplest of programs to advanced algorithms.

This document contains the information you need to know to program a standard TJ PRO™ robot. Other TJ PRO™ modules, such as the TJ PRO™ and the TJ PRO-ARGOS™, utilize extra libraries and modules. In order to program these robots, refer to the appropriate documents.

15.1 Overview of the TJ PRO™ Software Library

The TJ PRO™ software library is made up of a number of C source files, include files, and macro and symbol definitions. Almost any TJ PRO™ program you write will require the following include files and C files.

TJ PRO™ Include Files
- `hc11.h`   Symbol definitions specific to the HC11 (I/O ports, registers, etc.).
- `mil.h`   Macro definitions for simplified register manipulation.
- `tjpbase.h` Header file for all required TJ PRO™ header files.
            Must be in all TJ PRO™ C programs.
- `isrdecl.h` Header file for conditional declaration of all unused interrupts.
            Must be in all TJ PRO™ C programs.
- `analog.h` Header file for the analog function `analog.c`.
- `clocktjp.h`   Header file for the timing functions in `clocktjp.c`.

- `motortjp.h`   Header file for the motor functions `motortjp.c`.
- `serialtp.h`   Header file for the serial functions `serialtp.c`.
- `servotjp.h`   (Option)Header file for the servo motor control function `servotjp.c`.
- `vectors.h` Header file for interrupt vector specification `vectors.c`.

The header file `tjpbase.h` includes the last six header files in this list as a convenient way to include them with one include file. The header file `isrdecl.h` conditionally produces null functions for all unnamed interrupts. The `#define` at the beginning of that file lists the names defined interrupts.

TJ PRO™ Library Files
- `analog.c`   Routines for initializing and reading the analog ports.
- `clocktjp.c`   Timing functions, milliseconds, seconds, minutes, hours, days and a free running millisecond timer denoted by varialbe `timertjp`.
- `motortjp.c`   Drivers for PWM control of the wheel motors.
- `serialtp.c`   Serial routines for the robot to communicate to a terminal program on your PC.
- `servotjp.c`   (Option)Drivers and routines for controlling TJ PRO™'s servo motors.
- `vectors.c` Memory mapping of interrupt vectors.

15.2 Macro / Symbol Definitions

The files `hc11.h` and `mil.h` contain symbol and macro definitions that facilitate accessing and manipulating the HC11's memory-mapped registers. All register names are consistent with the names given in Motorola's *M68HC11 Reference Manual* (M68HC11RM/AD). For instance, to write to PORTB, you do not need to specify its memory address, instead you use the reference symbol PORTB. These two files assume that your chip's register base is at 0x1000. If you are not sure what this means, don't worry about it; if you've changed your register base for some reason, you'll need to modify the constant _IO_BASE in *hc11.h*.

Three macros that perform MC68HC11 register operations in ICC11 are defined in `mil.h`:

CLEAR_BIT(REG_NAME, MASK)
Will set to zero all the bits in REG_NAME which correspond to ones in MASK. Mathematically equivalent to: (REG_NAME *AND* (*NOT*(MASK))).

SET_BIT(REG_NAME, MASK)
Will set to one all the bits in REG_NAME which correspond to ones in MAKS. Equivalent to: (REG_NAME *OR* MASK)

CLEAR_FLAG(REG_NAME, MASK)
Will write a one to the bits in REG_NAME which correspond to ones in MASK. Generally, this is only used when clearing interrupt flags.

15.3 TJ PRO™ Defined Program Constants

Program defined constants permit you to refer to motors and sensors symbolically instead of trying to remember numbers associated with the different robot resources. The constants listed below can be found in the include file `<tjbase.h>` . You should use these symbols exclusively to be compatible with Mekatronix software. Note, that all defined constants consist of all capital letters. This is standard convention. Remember, the C compiler is case sensitive, so Bumper is not correct, but BUMPER is correct for specifying analog(0).

TJ PRO™ Defined Constants

```
/***************************** Constants *******************************/
#define LEFT_MOTOR   0
#define RIGHT_MOTOR  1
#define MAX_SPEED       100
#define ZERO_SPEED       0

#define BUMPER     analog(0)
#define RIGHT_IR   analog(2)
#define LEFT_IR    analog(3)

#define START  while(BUMPER<120)
#define FRONT_BUMP  (BUMPER>10)&&(BUMPER< 120)
#define BACK_BUMP  BUMPER>120

/* Enable OC4 interrupt and all servo operations */
#define SERVOS_ON     SET_BIT(TMSK1,0x10)

/*Disable OC4 interrupt: Stops all servo holding torques, useful for energy
savings*/
#define SERVOS_OFF    CLEAR_BIT(TMSK1, 0x10)

/*Turn IR emitters ON or OFF*/
#define IRE_ON  *(unsigned char *)(0x7000) = 0x07
#define IRE_OFF     *(unsigned char *)(0x7000) = 0x00

/********************** End of Constants *************************/
```

15.4 Analog Port Routines

To make use of TJ PRO™'s analog ports, you must include the file `analog.h` in your program. The following routines are available for controlling the analog ports. The C code for these routines are in `analog.c`:

init_analog()
Turns on the analog to digital converter. This function must be called once in your program before using any of the analog ports.

analog(int port)
Returns the value at the analog port corresponding to *port*.
Example: To get the value at analog port 3 into the variable "a", write: a = analog(3);

15.5 Servo Motor Routines

To use the servo motors (ARGOS option), include `servotjp.h` in your program. The corresponding C program is `servotjp.c`.

servo(int servo_name, int speed)
Will set one of the servos to a desired speed. To select which servo you want to move, two symbols have been defined: LEFT_MOTOR and RIGHT_MOTOR. The speed is given as a percentage of the motor's full speed.

Example 1: To move the right motor at half speed forward:

```
servo(RIGHT_MOTOR, 50);
```

Example 2: To move the left motor at full back speed:

```
servo(LEFT_MOTOR, -100);
```

head(int angle)
Will turn the head servo (TJ PRO-ARGOS™) to the angle specified. This angle ranges between –90 and 90 degrees. An angle of 90 will point the head to the right, -90 will point to the left, and zero points forward.

15.6 Writing your own Interrupt Service Routines for TJ PRO™ using ICC11

Only advanced users should try to modify `vectors.c`. If you have written an interrupt driver for an interrupt which is not already in use, you must open `vectors.c`, look for the name of the interrupt which you are planning to use, and change the default setting of 0x00 to the name of your interrupt service routine. You must, then, declare your service routine as an extern name at the top of `vectors.c`. You must also use #define <isr_name> at the beginning of the include file `isrdecl.h`.

You should not attempt to write interrupt service routines unless you have extensive experience with the MC68HC11 and have great familiarity with how interrupts work.

Mekatronix strongly advises you not to change existing interrupt service routines.

Caution!!!

If you change any of the standard TJ PRO™ drivers, Mekatronix applications and system software will not run correctly.

Use the example interrupt service routines provided in the directory `Libsrtjp` as guides to constructing your own. Be sure to `#define` your interrupt service routine name equivalent to the Mekatronix predefined names which are given for all possible MC68HC11 interrupt service routines in `Libsrtjp/vectors.c`. For those routines that you do construct be sure to enable installation of the interrupt vector by the compiler through a command sequence similar to the C-language construct in `servotjp.c`, namely,

```
#define servo_hand  TOC4_isr
#pragma interrupt_handler TOC4_isr
… … …

/*install interrupt handler on OC4*/
  *((void (**)())0xffe2)  = servo_hand;
```

A second, earlier, method which is not as "elegant", but maybe a little easier to understand, is illustrated by `motortjp.c` , first

```
#define motortjp_hand    TOC5_isr
…
…
…
#pragma interrupt_handler TOC5_isr
```

Next, to prevent the file `Include/isrdecl.h` from destroying the pointers created by the `#pragma`, you must add the lines

```
/* TJ Pro Motor handler */
#define TOC5_isr
```

at the beginning of `Include/isrtjp.h` . This `#defines` establish the existence of the interrupt service routine name, but assign nothing to that name (this has already been done!).   The conditional compilations that follow will recognize that this interrupt service routine name already exists and will not create a null function for it!

16. PROGRAMMING BEHAVIOURS

When you first get your TJ PRO™ we recommend that you write simple programs to become more familiar with the robot and its features while, at the same time, building your confidence in program development. Mekatronix also recommends the TJ PRO™ Education Package which includes a tutorial manual and a large number of source code programs illustrating various robot programming techniques.

 The next section suggests coding experiments to help you do just that.

16.1 TJ PRO™ Experiments

The numbered items below suggest a sequence of ever more complex programs and experiments you can perform to familiarize yourself with TJ PRO™'s capabilities. These experiments will open up to you the rich variety of behaviors you can program. After each successful experiment, save your program and do not change it. Use copies of it to begin other programs, but do not write over and destroy your only copy of a successful program. You will never know when you might want to use it again, either as is, or as a basis for another program.

16.1.1 Robot Experiments

1. **Motivation**

    TJ PRO™ motion control depends upon being able to start the motors.

    Objective

    Learn to code a simple program to turn on one of TJ PRO™'s motors.

    Specification

    Turn on a motor 100 percent in the forward direction. Modify this program to turn on a motor 100 percent in the reverse direction.

    Questions

    Did the motor turn in the direction you thought it would when you coded it?

2. **Motivation**

    Robots often use dead reckoning to navigate and make maps in non-critical situations. The ability to go straight enhances the accuracy of dead reckoning.

    Objective

    Move the robot forward and determine if it goes straight.

    Specification

    Turn on both motors in the forward direction at the same percentage.

    Questions

    Does the robot go straight? Which way does it prefer to turn, to the left or to the right? Which motor, left or right, appears to be turning faster for the 100% forward command?

3. **Motivation**

    DC motors may not have symmetrical characteristics. Their performance may differ depending on the direction they turn.

    Objective

    Move the robot in reverse and determine if it goes straight.

    Specification

    Turn on both motors in the reverse direction at the same percentage.

    Questions

    Does the robot go straight backwards? Which way does it prefer to turn, to the left or to the right? Which motor, left or right, appears to be turning faster for the 100% reverse command? Are your results consistent with Experiment 2?

4. **Motivation**

Quantification of the deviation from straight-line motion may lead to compensation techniques to improve that motion.

Objective

Measure the robot's deviation from straight-line motion when it is supposed to be going straight.

Specification

Turn on both motors at 100% forward for 10 seconds and stop the robot.

Procedure

This experiment can be performed easily on a tile floor. Just line up the left wheel on a tile line and determine how far the wheel has deviated from the axis of that tile line after the robot stops. Extend the wheel axis at the start and stop positions with a string. Assuming the robot is turning in a big circle, the intersection of the two strings determines the radius of the circle(*Why?*).

Questions

After 10 seconds, how many inches has the robot moved forward along its initial direction? How many inches has the robot veered from the line of its initial direction?

5. **Motivation**

A robot often spins as a preliminary maneuver to escaping a difficult situation.

Objective

Learn to spin TJ PRO™ in various ways for a fixed length of time.

Specification

Make the robot spin clockwise about the left or right wheel. Modify your program to make TJ PRO™ spin about its center axis.

Procedure

At the beginning of the experiment, place erasable marks on a tile floor at the outside edge of each wheel, just below the center of the axle. Draw a line between the two marks. The center of this line is the center axis of the robot. At the end of the experiment repeat the marking procedure. Measure the distance and the compass heading between the center of the two lines you have drawn.

Questions

For each of the two types of spins, did the robot stay in one place or did it drift? Measure the drift as described in the procedure. Can you explain what you saw?

The experiments listed above will help you understand some of TJ PRO™'s motion characteristics. You can devise others to test TJ PRO™'s sensor capabilities.

16.2 Scope

Programming behaviors is what autonomous mobile robots is all about, or, at least a substantial part of what it is all about! Without being technical, a behavior is whatever the robot does. The emphasis is on action! From the engineering viewpoint, you want to program behaviors that produce useful results. Make a robot vacuum cleaner, or a robot valet. With an artistic eye, you want to program behaviors that esthetically please or excite. Why not make TJ PRO™ dance? A

ballet on wheels! From the scientific perspective you can inquire about the scope of machine intelligence and test your theories on a real robot! Out of intellectual curiosity and the creation urge, you might want to develop physically embodied animats, artificial animals. Develop your own ecology with predator robots that drain the prey robots' batteries and prey robots that hide and avoid predator robots and seek battery recharging stations as food sources. Or, you can tailor a robot to enter many of the robot contests around the world. Many of these contests require manipulation and sensors not supplied with TJ PRO™. But, with one or more MSCC11 single chip computers controlling the additional sensors and servo driven manipulation devices, a TJ PRO™ can often be expanded to meet contest requirements.

16.3 Some Possible Behaviors

TJ PRO™ programs provide the basic hardware interrupt and device driver routines for the robot. These allow the user to access the sensor readings and drive the motors and servos. With these routines, the user can program an unlimited number of behaviors. A representative set, but, by no means, an exhaustive set, of primitive set of behaviors, from which more complex ones can be developed, are listed in Table 2.

Table 2 Primitive Behaviors

TJ PRO™	TJ PRO™	TJ PRO-ARGOS™	TJ PRO-ARGOS™
Collision Avoidance	Collision Detection	Line following	Light following
IR Light avoidance	Pushing	Collision detection	Shy behavior
Aggressive behavior	Exploring behavior	Wall following	Beacon tracking

16.4 Advice on Developing Behaviors

The following advice is based on several years experience teaching engineering students to program autonomous robot behaviors.

16.4.1 Vulcan Mind Meld

To effectively program a behavior for TJ PRO™, or, quite possibly, any autonomous robot, and to gain insight into the problems encountered by your robot, you should play Vulcan to the robot and imagine performing a Vulcan mind meld with it. All you Trekkie fans know what this means. But, to be specific, try to perceive the universe as the robot does with its limited capabilities. As you imagine yourself one with the robot, play out different sensations and responses. Help yourself by actually recording robot sense data and examine typical responses, or responses to special environmental conditions of interest in the behavior you are developing. The mind meld will help prevent the common error of asking the robot to respond to environmental conditions it cannot detect with its sensors! While this statement is so totally obvious, it is also a difficult self-discipline to psychologically enforce. Why? Humans typically interact with each other or intelligent animals, expecting and perceiving sophisticated behavior and sensory performance. These expectations seem to subconsciously creep into our agenda when working with autonomous machines, often with disappointing results! Autonomous robots have no where near the sensory and behavioral capabilities of an insect, let alone higher animals.

16.4.2 Virtual Mind Meld©[4]

To assist in perceiving the universe as the robot does, you can write programs to generate computer graphic displays that depict the robot's perception in any sense that makes communication with the robot easier. Robot Rorschach tests, color maps…a virtual robot environment. This process we coin as a *Virtual Mind Meld.* The robot portrays its reality in the computer graphics medium to create a virtual reality to bridge the species communication barrier.

16.4.3 Relative calibration of sensors of the same type

Manufacturing tolerances, circuit tolerances, and mounting variations make it possible for two instances of the same type of sensor to respond differently to the same stimulus. Behaviors, therefore, should not be programmed to depend upon two sensors of the same type producing identical responses to the same stimulus. Instead, program sensors of the same type to relatively calibrate themselves in some fixed environment. For example, place a cardboard box in front of, and parallel to, the wheel axis of a TJ PRO™. Measure the response of the two front IR detectors. Note the differences in the readings. If there are none, that's great! In general, however, they will differ somewhat.

Program the robot's behaviors to respond to relative sense stimuli, not absolute sense measurements. This will make the robot behave more organically and robustly to uncertain, dynamic environments.

16.4.4 Adjusting to Ambient Conditions

A programmed behavior will often be *brittle*, not flexible or adaptive, if that behavior depends upon specific magnitudes of robot sensor readings. Brittle behaviors fail when the environment changes from the environment in which the behavior was developed. The smaller the change that causes the failure, the more brittle that behaviors is. For example, the IR detectors on TJ PRO™ will detect white objects at larger distances than dark objects. Suppose a collision avoidance algorithm sets a threshold value of the IR as an indication of an impending collision. If this threshold is determined experimentally with light colored obstacles, then dark colored obstacles will not be detected and the robot will bump into them. On the other hand, if the threshold is set for dark colored obstacles, the robot will end up spinning in circles in a light colored environment because it detects threats everywhere. The solution is not to pick an average color threshold, but rather, program the robot to adjust its threshold downward if it has not detected a collision for some specified time, or, to adjust the threshold upward if it is colliding too frequently. The difficulty, of course, is determining exactly what the "specified time" between collision should be or what "colliding too frequently" means! The easy, but difficult to implement, answer is to let the robot learn these parameters based upon some performance criteria.

Robot behaviors and sensors, therefore, should adjust to ambient conditions. Biological organisms perform this function fantastically well. The human eye adjusts to bright sunlight or a darkened

---

[4] Virtual Mind Meld ©1997 by Mekatronix™

cathedral with dimly lit candles. This procedure is easier to state then execute, but serves as a general principle.

16.4.5 Create simple behaviors

The beginning robot practitioner usually formulates behaviors too complicated to implement directly. With experience, the virtue of simple, direct behaviors becomes apparent. Complex behaviors should be broken down into sequences of simple, primitive behaviors. If this can be done, the chances of successful implementation are high. If not, there is little value in trying to implement such behaviors directly.

16.4.6 Build on simple behaviors

As the user accumulates a repertoire of primitive behaviors, complex behaviors open up. Perhaps the easiest way to generate complex behaviors is simply to sequence a collection of primitive behaviors. For example, wall following might be decomposed as follows: 1) detect a "large" object, 2) approach the object until "near", 3) turn until the robot front-to-rear axis aligns "parallel" with the "surface" of the obstacle, 4) move "parallel" to the obstacle surface. At each instant of time a particular behavior in the sequence is invoked based on the current state of the robot and its sensory inputs. Of course, the programmer will have to establish to the robot's perception the meaning of such terms as "large", "near", "surface", and "parallel". Remember to Vulcan Mind Meld!

16.5 Integrating Behaviors

More complex behaviors may require the combination of primitive behaviors in a way not well understood. Artificial neural network activation, opinion guided reaction, non-linear dynamics, and fuzzy logic all offer techniques for integrating behaviors. Each technique offers specific advantages and specific difficulties. Discussion of such issues is beyond the scope of this manual. The reader's attention is brought to this matter to encourage investigation into these possibilities.

17. TJ PRO™ TROUBLESHOOTING GUIDE

To check out your robot, download the program file `avoidtjp.s19` in the `tjpcode` directory in the distribution software and run it.

Note. *If the robot moves, avoids obstacles, and responds to a bumper switch closure when running Avoidtjp.s19, all hardware systems work and your problem must be software. If TJ PRO™ does not avoid obstacles and does not respond to bump contact, use the following trouble guide.*

Is the Problem Hardware or is the Problem Software?
Engineers point their finger at the software and the programmers fault the hardware! The reality, unfortunately, is that determining the causes of an error or malfunction can be frustratingly difficult. Problems can arise from either or both sources. When you encounter an intractable debugging problem, the fault almost invariably stems from a false assumption about the working state of your robot and your program. Systematic testing of both hardware and software, on an

incremental basis, can greatly reduce errors and help you to isolate causes of errors when they do occur…and they will occur, that is a promise!

Note. In the troubleshooting guide below, you should always keep in mind that problems may have multiple causes and only some highly probable ones are mentioned.

Check Battery Voltage
Many hardware and apparent software problems with your TJ PRO™ robot result from uncharged or low batteries. Under low battery conditions, loaded programs often become corrupt or the robot processor resets every time the motors pull large amounts of current from the batteries, for example, when the motors make substantial changes in speed or direction, as in starting and stopping.

Symptoms of Low Battery Voltage
As the batteries become low the red power LED blinks when the motors pull large currents. This warning should be heeded and the batteries charged or exchanged for fresh ones. A more radical symptom occurs when the batteries drain too low to support current demands of starting and stopping the motors. The processor rapidly resets and restarts your program each time the motors demand current. In such cases, TJ PRO™ stutter-rolls forward and does little else.

IMPORTANT! When troubleshooting first check to make sure that your batteries are fully charged and the red power LED lights when the On/Off switch is flipped to On.

17.1 IC Trouble Guide
As long as your code is syntactically correct and IC downloads, then errors can usually be attributed to your code itself. The most common problem is not being able to get IC to talk to your robot. This is typically a COM problem, an issue discussed in both Sections 13 and 14.

17.2 ICC11 Trouble Guide
The numbered items below indicate the most common problems encountered their possible causes and associated fixes.

1. **My program does not compile**. *Refer to Section 14 and redo the instructions appropriate for your case. Be sure you execute iccsetup.bat each time you open a new DOS Window or start up in DOS. If that does not work, you might want to repeat the whole installation procedure in Section 14.*

2. **My program compiles with errors**. *Here you must use your program debugging skills. Program debugging is beyond the scope of this manual.*

3. **My program compiles, but does not download**. *Refer to Section 14.5.*
    3.1. *Be sure the Download/Run switch is in the Download position.*

3.2. *Verify that the robot connected to the right COM port.*

3.3. *Check and verify all the physical connections between your PC, the Communications Board (MB2325) and the TJ PRO™. Diode D1 should be on when the Com-board is connected with your PC. LED D2 should light when the reset button on TJ PRO™ is held down.*

17.3 General Trouble-Shooting.

4. **My program downloads, but nothing happens when I press the reset button.**

   4.1. *Check to make sure you have switched the Download/Run switch to Run and that the power switch is On.*

   4.2. *Press the reset button several times. Sometimes the switch does not make proper connection.*

   4.3. *Your program may actually be running, but program logic errors prevent any observable robot behavior. This is a tough one. Download and run a working program such as* `avoidtjp.s19` *to verify all TJ PRO™'s hardware functions. Also, download* `sensetjp.s19` *to test all the sensors. If you are successful, then the problem must be in your program.*

   4.4. *If you cannot successfully run* `avoidtjp.s19` *or* `sensetjp.s19,` *then unplug the battery pack and check the reset button with an multi-meter for switch closure (short circuit) when the button is pressed. Replace the switch if it does not operate properly.*

   4.5. *If the switch works, check continuity of the wiring to the reset button and repair if necessary.*

5. **My program downloads, but when I press the reset button the robot just jerks and stutters.**

   5.1. *Check for weak batteries. Weak batteries can cause the processor to reset every time the motors demand current. Weak batteries can also corrupt your program. Replace batteries with fresh ones or recharge batteries.*

   5.2. *Your program may be syntactically correct but possesses logic errors, causing the robot to behave radically. If you believe your code to be correct, download and run* `avoidtjp.s19` *to verify, or not, all TJ PRO™'s hardware. If the hardware checks out, the problem is in your program.*

6. **Power LED Does Not Light when the *On/Off* switch is flipped to *On*.**

   6.1. *Batteries are not charged. Recharge or replace with fresh ones.*

   6.2. *Battery cable from the battery to the top plate is loose. Reseat connector.*

   6.3. *Battery cable from the battery to the top plate is reversed. Turn connector 180 degrees and reseat securely.*

   6.4. *Battery snap connector is loose. Resnap.*

   6.5. *Batteries have vibrated loose and are not making good contact. Reseat batteries.*

   6.6. *One or more of the wires to the battery connector have broken away from the connector pin. Resolder the wire(s) to the connector and reseat connector.*

   6.7. *One or more wires in the power circuit involving the LED have broken loose. Resolder.*

If none of the above work, but you can download programs and run the robot, then the power LED must be replaced.

7. **My program downloads, but when I press the reset button the robot backs up and turns repeatedly.**
    7.1. *The bumper has probably jammed a bumper switch closed. Adjust the bumper so that all switches clear the bumper.*
    7.2. *If the bumper clears the switches and the behavior persists, check your code for program logic errors. If you believe your code to be correct, download and run* `avoidtjp.s19` *to verify, or not, all TJ PRO™'s hardware. If the hardware checks out, the problem is in your program.*

8. **My program runs, but the robot always bumps into things.** *Possible sources of trouble: programming error, IR emitter and detector connections, IR LED wired in reverse biased. Run sensetjp.s19 to check operation of the sensors. If IR does not respond to a white sheet of paper moved to and from the robot, then*
    8.1. *Verify that all IR connections are secure and properly oriented and no wires have broken.*
    8.2. *If you have a monitor driven by a black-and-white CCD camera without an IR filter, you can verify that you have IR by pointing the emitters at the camera. The IR will appear as visible light in the image. You can easily see if flipping the IR emitter connector forward biases the diode because it will begin to shine. You can also use another, working TJ PRO™ to make this test!*

9. **Motors don't turn.**
    9.1. *Did your program execute* `init_motors()?`.
If only a single motor doesn't turn, then
    9.2. *Verify the motor plug is connected to the correct header and is not reversed.*

10. **Motors turn the wrong direction.**
    10.1. *Possibly a wrong motor specification in your program.*
    10.2. *Verify the left motor plug is mated with the left motor header. If not reverse motor plugs.*

Comments on the IR System

IR Detector Connectors: Black wire fits the pin next to the edge of the can
We have intentionally connected the IR detectors incorrectly and have not observed any damage to the detectors. Of course, they do not function when connected incorrectly. However, we recommend avoid doing so.

IR Emitters Shine
How can you tell if the IR emitters emit IR? Run the program `avoidtjp` and see if TJ PRO™ avoids obstacles placed on either side of the robot. You can also run `sensetjp` and check

operation of the IR detectors. If the detectors change values as you bring your hand towards the robot, the IR emitters are working.

18. TJ PRO™ GENERAL DESCRIPTION

While the TJ PRO™ JUNIOR PROFESSIONAL™ (TJ PRO™) shares sibling DNA with TJ™, the TJ PRO™ has more features and capabilities. A single Mekatronix microcontoller board, the MTJPRO11, provides all the computational, sensing and control need for sophisticated robot behaviors. The following paragraphs provide a brief description of the TJ PRO™ technical characteristics.

18.1 Mechanical Structure

1. All of TJ PRO™'s body parts are made from either beautiful, strong, durable, 1/8 inch thick, 5-ply, model airplane plywood or from matte black ABS plastic.
2. TJ PRO™ fits into a right circular cylinder 7 inches in diameter by 3.25 inches high. (Volume approximately 125 cubic inches or 0.072 cubic feet)

18.2 Power Requirements

1. Six AA rechargeable Nickel-Cadmium batteries with at least 600 ma-hr capacity, 5.4-7.2 volts (Sold separately). Premium batteries at 700 ma-hr capacity provide more run-time for your robot.

WARNING!
USE ONLY NiCd BATTERIES FOR TJ PRO™. DO NOT USE ALKALINE OR OTHER BATTERY TYPES WHICH WILL DESTROY THE ROBOT ELECTRONICS.

2. Recharger, 12 volts D.C. rated at 500ma , must be at least 200ma. (Sold separately).

18.3 Actuation

TJ PRO™ can control up to four servos, either hacked as gearhead D.C. motors or as servos. Two motor control ports are reserved for the wheel motors and two will be used for a pan tilt head on the ARGOS enhancement kit (optional). No servo should be attached to OC4 because IC will burn it out.

        Gearhead DC motor drive for each wheel.
1. 5.4- 7.2 Volts.
2. 100 -120 ma under load, 80 ma no-load.
3. 1.25 revolutions/sec at 7.2 volts (full battery charge). Speed decreases proportionally to the voltage as it drops.

18.4 Robot Controller

A Motorola MC68HC11 based microcontroller circuit, the MTJPRO11, provides, among other features, the brains and communication capability of the TJ PRO™ and provides exceptional versatility for the robot. Control and communication features include the following.

1. MC68HC11 microcontroller.
2. 32Kbytes of SRAM.
3. Five-volt serial communication interface for downloading and uploading programs and data to a personal computer. The serial bus implements RS-232 protocol except for the voltages. The Mekatronix circuit board MB2325 converts this processor serial bus to a standard RS-232 bus.
4. High speed, 5volt synchronous serial bus. Up to 1 MHz data rate.
5. 5 Volt regulator.
6. Low-voltage-inhibit reset circuit.

18.5 Memory Save Feature

When the power switch is turned off, power is still supplied to the SRAM to prevent loss of code and data in memory. The batteries will be drained after about a day if you do not keep TJ PRO™ plugged into the charger unit.

Recommended.
 When not in use for more than a few minutes, plug TJ PRO™ into the charger unit to keep the batteries fresh. During prolonged programming sessions, Mekatronix also advises you to keep TJ PRO™ connected to the charger so that the robot batteries will always have a full charge during experimental runs.

18.6 Input-Output Expansion Capability

The MTJPRO11 circuit board provides TJ PRO™ with exceptional IO power for a robot its size (Mekatronix will offer future enhancement kits based on these capabilities. Check the web site for further details [http://www.mekatronix.com]).

1. Eight digital outputs.
2. Three digital input ports with program-optional automatic digital waveform capture.
3. Hardware generation of a 40KHz signal to modulate IR and sonar.
4. The processor data bus and an 8-bit address decoder port enable expansion up to four external 8-bit Digital Input and four external 8-bit Digital Output ports.
5. Five, eight-bit analog input channels available. Combined with external analog multiplexers and Digital Input address decoding, each analog channel could be multiplexed to as many channels as the external multiplexer allows.

18.7 Memory and IO Mapping for the MTJPRO11™

The 32Kbyte SRAM occupies the upper 32Kbytes of the MC68HC11 address space, namely, from hexadecimal 0x8000 to 0xffff. The MTJPRO11™ provides four, active-low Input-Enables Y1, Y3, Y5 and Y7 and four, active-low Output-Enables Y0, Y2, Y4, Y6. These enables map into memory address space as listed in Table 3. A read of the appropriate input memory address enables that input. Similarly, a write to the appropriate output memory address enables that output. For example, a read of memory location 0x5000 actually enables Y3 and reads the data bus. Suppose an 8-bit register attached to the data bus outputs its contents onto the computer data bus when enabled by Y3. Effectively, a read of 0x5000 will read the contents of that register.

Gainesville, Florida   www.mekatronix.com   Tech Questions: tech@mekatronix.com

Such a register is an 8-bit input port. A write to the same address enables the output port selected by Y2. Suppose Y2 enables the inputs of an 8-bit register attached to the computer data bus. The write operation will store the contents of the computer data bus into that register. The Y2 output port does not have to be the same register as the Y3 input port. The choice is up to you.

Table 3 Memory Map of MTJPRO11 IO Enables

Name	Direction	Memory Address (Hex)
Y0	Output	0x4000
Y1	Input	0x4000
Y2	Output	0x5000
Y3	Input	0x5000
Y4	Output	0x6000
Y5	Input	0x6000
Y6	Output	0x7000
Y7	Input	0x7000

18.8 TJ PRO™'s Starter Sensor Suite

Every TJ PRO™ comes with the minimum sensor suite listed in Table 1. These sensors are wired directly to headers on MTJPRO11 microcomputer circuit board.

These sensors have the following characteristics:

1. Two Forward-Looking IR Emitters, wavelength equals 940nm.
2. One Backward-Looking IR Emitter, wavelength equals 940nm.
3. Two Forward-Looking analog IR Detectors for 40KHz modulated 940nm IR. These sensors produce analog channel readings from about 88 to 128 out of a possible 256. The number 256 corresponds to five volts.
4. Three Front bumper Momentary Tactile Switches, each switch closure separately identifiable.
5. One Back Bumper Momentary Tactile Switch
6. User expandable sensors (see Section 18.5 for IO expansion capabilities).

Table 4 TJ PRO™'s Sensor Suite

TJ PRO™ Label	Name	Function
IRDLF	Infrared Detector, Left Front	Proximity Sensor
IRDRF	Infrared Detector, Right Front	Proximity Sensor
FBLSW	Front Bumper Left Switch	Front contact Sense
FBCSW	Front Bumper Center Switch	Front contact Sense
FBRSW	Front Bumper Right Switch	Front contact Sense

RBSW	Rear Bumper Switch	Rear contact Sense

18.9 Switches

1. Reset push button
2. Toggle switch: Download Program and Run Program
3. Toggle switch: Off-On.

18.10 System Support Software

TJ PRO™ programs can be written in MC68HC11 Assembly Language, C, or BASIC.

1. Sensor and motor routines provided in assembly language.
2. A freeware version of the Interactive C (IC), an extremely useful C interpreter by Randy Sargent. Ask your Mekatronix distributors about the up-to-date commercial version of IC.
3. PCBUG11 freeware for downloading Motorola S19 files.
4. Freeware version of Basic for programming TJ PRO™.
5. Freeware MC68HC11 Assembly Language.

Separate purchase of a commercial C compiler is also available. Contact us for more information

18.11 Serial Communication

To develop your applications requires communications between a Personal Computer and TJ PRO™. The additional purchase of an MB2325 communications board and a 6-wire RS-232C communications cable will provide the hardware for that capability. Only one MB2325 board and cable is necessary to enable you to sequentially load and download any number of Mekatronix™ robots, since the MB2325 board can remain attached to the PC and not the robot.

ENJOY YOUR TJ PRO™ ROBOT and tell us about your TJ PRO™ robot rexperiences!