Universal Programmable Robot Controller

with motor driver

Hardware Manual Rev 1r1

Z8 Encore!XP® MICROCONTROLLER ENGINE

The Universal Programmable Robot Controller (**UPRC)** uses feature rich Z8F082A Zilog Encore! XP microcontroller engine and inherits its features as follows:

Internal Precision Oscillator (IPO). The Z8F082A has built-in clock source, eliminating the need for an external crystal or oscillator. Two selectable IPO frequencies are available, a 32.8 kHz and a 5.53MHz clock source. 32.8 kHz is usually selected during standby conditions to keep power consumption low, a very important feature with battery operated or battery back-up products.

20MHz maximum clock speed using external crystal. The internal 5.53 MHz clock speed is comfortably fast enough for most applications. In those very few instances when speed becomes crucial, you can easily turbo charge the Z8F082A up to a blazing 20MHz speed by using an external crystal. The UPRC has a socket reserved for this purpose. Just plug a crystal in, throw in a couple of loading capacitors, and your Encore! revs up to speed. The only downside is the crystal connection takes two I/O ports for its operation.

On-Chip Debugger OCD and Flash Programmer. Use your PC to program and debug the UPRC, in-circuit. You need only a debugging cable, a.k.a dongle. No need for a separate (and expensive) programming device. Its built-in OCD, allows you to examine and manipulate the chip memory, trace program flow, assign breakpoints, and lot of other debugging tasks that previously you can do only with the aid of a dedicated emulator board.

8K flash memory with 10,000 writes endurance. Experienced microcontroller users knows that 8K program memory is big enough even for complex control tasks, as long as you stay away from memory hogging functions like C's printf(). 10,000 times write endurance means you can repeatedly change the program in the flash ROM at least 10,000 times before the chip gives up.

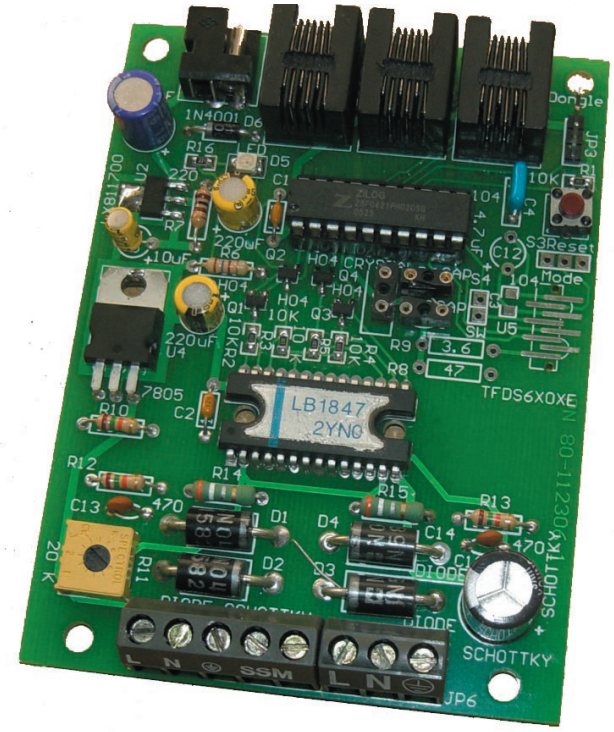


Figure 1. The Universal Programmable Robot Controller module uses feature rich Zilog Encore! XP MCU with in-circuit programmable Flash memory, and sports a motor driver subsystem that can drive two DC motors or one stepping motor.

Encore! and Encore! XP are registered trademarks of Zilog , Inc.

UPRC is designed by Engr. Rodolfo De Guzman
Manual written by Henry Chua

Copyright 2007 by e-Gizmo Mechatronix Central
All rights reserved.

No part of this publication may be reproduced in any form without the written consent of
e-Gizmo Mechatronix.
Content subject to change without prior notice.
All informations contained herein are believed to be correct and reliable.

Before using this document, you must agree with the following terms and conditions:

1. e-Gizmo Mechatronix and the author cannot be held liable for any damage that may occur with the use or misuse of any information contained in this document.
2. You are allowed to reproduce this publication and the product it describes for personal use only. Commercial reproduction is prohibited!

1K SRAM. Plenty of RAM space for all your temporary data. Being an old hand used to working with z8 OTP microcontrollers with only 256 bytes of SRAM memory, I find 1K SRAM an awesome feature!

The Analog to Digital Converter (ADC) is the controller communications medium with the analog world. With 10 bit resolution to boot, the ADC can measure analog input changes as little as 0.001V. You can use these inputs to measure and process the analog components of your system. The UPRC has 3 input channels available for user applications.

Two 16-bit multifunction timers. This is one of the most useful built-in peripherals of the Encore! Microcontroller. You can use the timers to generate time delays, measure time, frequency, generate pulse and pulse width modulation (PWM) functions, count events, and so on.

Universal Asynchronous Receiver-Transmitter (UART) gives your controller the ability to talk with other similarly equipped controllers or equipment. For example, you might find the need to use more than one UPRC controller in your system. It is essential for this kind of setup for one UPRC to know what the other UPRC is doing, and of course, you can conveniently set up their private hotline via the UART channels. Another example is, with the aid of RS-232 interface chip (e.g. MAX232), you can connect your UPRC to a PC.

On-chip Temperature Sensor. Lets you know the chip internal temperature. With some clever techniques, you can use the sensor as well to measure the temperature surrounding the chip.

On-chip Analog Comparator. Gives you another way to work with analog inputs. This circuit will tell the controller if the measured input is above or below a preset analog voltage.

IRDA Encoder/decoder. The UPRC has an optional IRDA circuit that will allow you to communicate and exchange data with similarly equipped devices.

ZDS II Development Environment. Zilog generously made this C and Assembly language development software free to zilog Encore! Users. The download package includes comprehensive manuals and sample programs. Visit Zilog site www.zilog.com for details.

UPRC Motor Driver

It has a LB1847 PWM Dual H bridge Motor driver built-in, making it easy for you to connect and control motors:

- Can be used to drive two DC motors independently (Forward, Reverse, OFF)

- Can be used to drive one stepping motor in PWM current mode

The motor driver has adjustable current limit. It will switch to PWM mode once the maximum preset current is reached to keep the current more or less equal to preset value even with increasing load. It is also over-temperature protected, if it runs too hot, it will shut itself OFF.

IMPORTANT REMINDERS

UPRC is ESD sensitive. Observe the following handling precautions to prevent damage.

- Keep yourself connected to earth via ground wrist strap.
- Never work on plastic furniture.
- Keep your table free from unnecessary clutter.
- Keep your components inside ESD shielding containers until the moment these are installed. Do not store your components inside ordinary plastic bags and containers.

More on ESD :
http://www.e-gizmo.com/ARTICLES/ESD/ESD1.HTM

Heatsink may be required on IC LB1847 if the motor driver is operated with heavy loads. Indication that heatsink is needed is when LB1847 runs too hot to touch during operation, or if the driver shuts down.

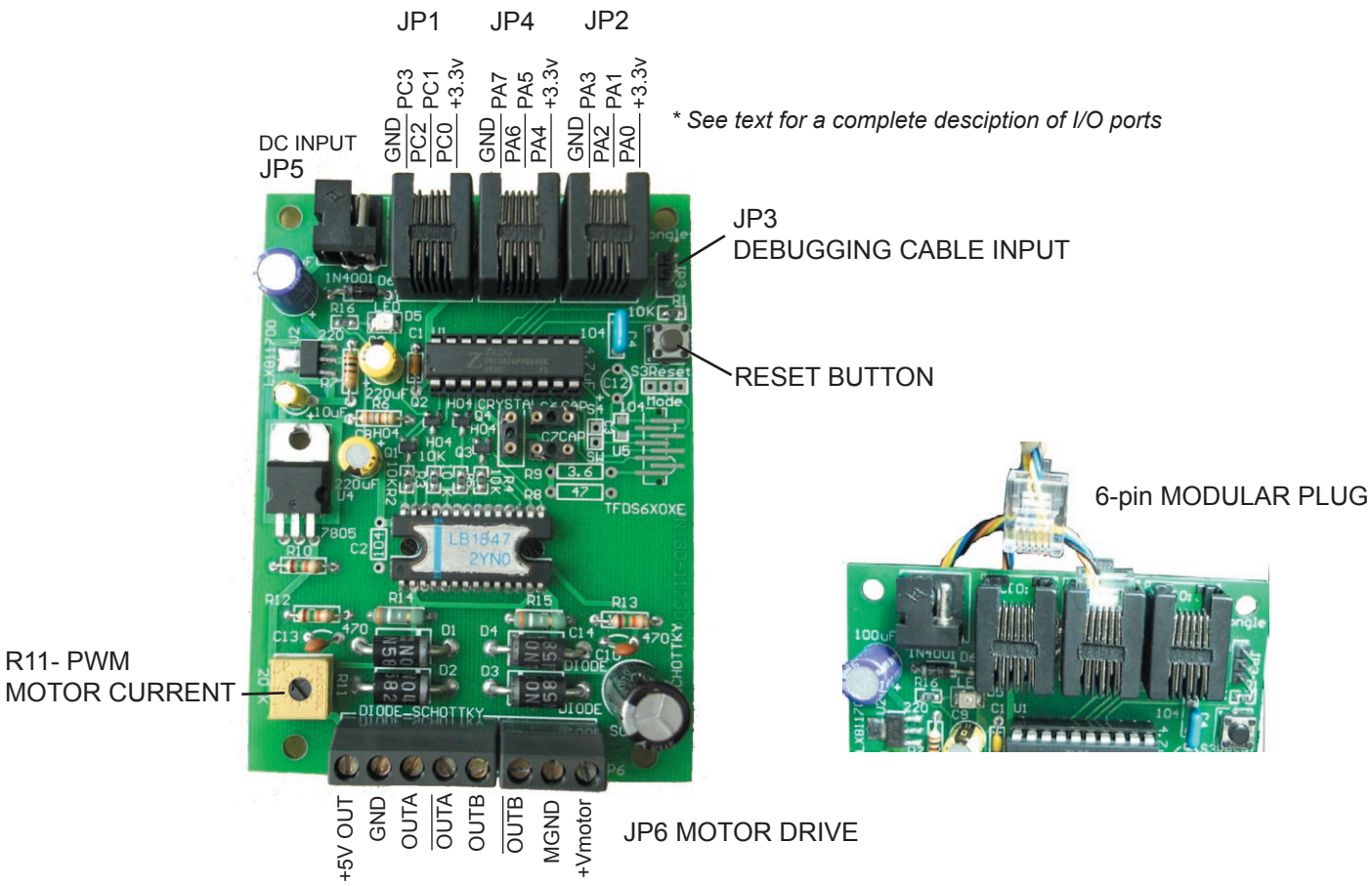


Figure 2. UPRC has 12 I/Os available for user applications. The I/Os are terminated by modular connectors, making access to these ports fast and easy. Screw terminals are used for the motor drive circuit because of the relatively large amount of current passing through these points.

PIN DESCIPTION

The Z8F082A Encore! XP has plenty of on-chip peripherals. Making these peripherals available to users on a DIP package with only 20 pins obviously presents a design challenge. Zilog's solution is for each peripheral to share pins with the others. Hence in the following pin descriptions, you will find most of the pins having two, three, or even four possible functions. Take JP1 pin 2 pin as an example. It is labeled as PC0/ANA4/CINP/LED. This means it can be configured take the function of a logic input or output PC0, or an ADC analog input ANA4, or a comparator input CINP, or a LED direct drive output PC0. This scheme, unfortunately, prevents us from using all available peripherals at the same time.

JP3 – Dongle (Programming Cable)

Pin 1 : +3.3V
Pin 2 :GND
Pin 3 : OCD – On chip Debugger pin, used by ZDS II PC development environment to program and debug and program the chip in-circuit.

JP1 – User I/O

Pin 1: +3.3V
Pin 2: PC0/ANA4/CINP/LED
Pin 3: PC1/ANA5/CINN/LED
Pin 4: PC2/ANA6/LED
Pin 5: PC3/COUT/LED
Pin 6: GND

© 2007 e-Gizmo Mechatronix Central

PC0.. PC3
General purpose logic input/output
ANA4..ANA6
10-bit ADC analog inputs
CINP
Comparator (+) input
CINN
Comparator (-) input
COUT
Comparator Output
LED
Direct LED drive capable outputs

JP2 – User I/O

Pin 1: +3.3V
Pin 2: PA0/T0IN/T0OUT/XIN
Pin 3: PA1/T0OUT/XOUT
Pin 4: PA2/DE0
Pin 5: PA3/CTS0
Pin 6: GND

PA0..PA3
General purpose logic input/output, PA2, PA3 5V tolerant (note 1)
T0IN
Timer0 input. This input can be used for gating, counting, and capture functions
T0OUT
Timer0 output.
$\overline{\text{T0OUT}}$
Complement of T0OUT in PWM Mode.
DE0
Device Enable. This allows automatic control of external RS-485 drivers.
$\overline{\text{CTS0}}$
Clear to Send , flow control input for the UART.
XIN
External crystal input.
XOUT
External crystal output.

JP4 – User I/O

Pin 1: +3.3V
Pin 2: PA4/RXD0
Pin 3: PA5/TXD0
Pin 4: PA6/T1IN/ $\overline{\text{T1OUT}}$
Pin 5: PA7/T1OUT
Pin 6: GND

PA4..PA7
General purpose logic input/output, 5V tolerant (note 1)
RXD0
Receive input of UART or IRDA
TXD0
Transmit ouput of UART or IRDA
T1IN
Timer1 input. This input can be used for gating, counting, and capture functions
T1OUT
Timer1 output.
$\overline{\text{T1OUT}}$
Complement of T1OUT in PWM Mode.

Note 1: Will work with 5V logic inputs when configured as inputs with pull-ups disabled.

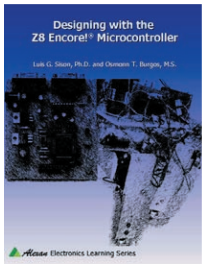
JP6 - Motor Driver Terminal

Pin 1: +5V Output
Pin 2: GND
Pin 3: OUTA
Pin 4: $\overline{\text{OUTA}}$
Pin 5: OUTB
Pin 6: $\overline{\text{OUTB}}$
Pin 7: MGND
Pin 8: Vmotor (+)

+5V Output
+5V 300mA power source for add-ons modules.
GND
Logic Ground.
OUTA, $\overline{\text{OUTA}}$
Complementary Drive Output A
OUTB, $\overline{\text{OUTB}}$
Complementary Drive Output B
MGND
Motor Supply GND
Vmotor
Motor Power Supply input (+)

USING THE UNIVERSAL PROGRAMMABLE ROBOT CONTROLLER (UPRC)

This guide assumes that you are already familiar with Z8 Encore! Microcontroller chip. Detailed discussion of the chip's architecture and programming is beyond the scope of this document. Zilog's site www.zilog.com contains a wealth of information detailing everything you need to know about this chip, from specifications to application examples.



Highly recommended readings is a book entitled *Designing with Z8 Encore! Microcontroller*, written by authors Dr. Luis Sison and Prof. Osmonn Burgos of U.P. Diliman. This book is written in tutorial style and contains numerous exercises that will make it easy for you to learn and master the Encore! chip. This book is published by Alexan Commercial and is available through Alexan chain of stores.

Connecting a Power Supply

The UPRC requires two power supply, one that powers the controller circuit, and a separate one that powers the motor driver circuit. You can operate the controller without the motor driver power supply if you do not intend to use the motor circuit.

Power to the controller is supplied via the DC jack. You can use power supply or battery voltage ranging from 7.5V to 9V DC. Current consumption is low, it typically drains less than 100mA, but actual consumptions may vary depending on the gadgets that you connect with the board.

The motor driver can accept DC power input up to 24VDC. If you are using the driver circuit to control DC motors, you should use a supply voltage slightly higher than the motor rating to compensate for losses on the driver circuitry. If you are to drive a stepping motor in PWM mode, then use 24VDC.

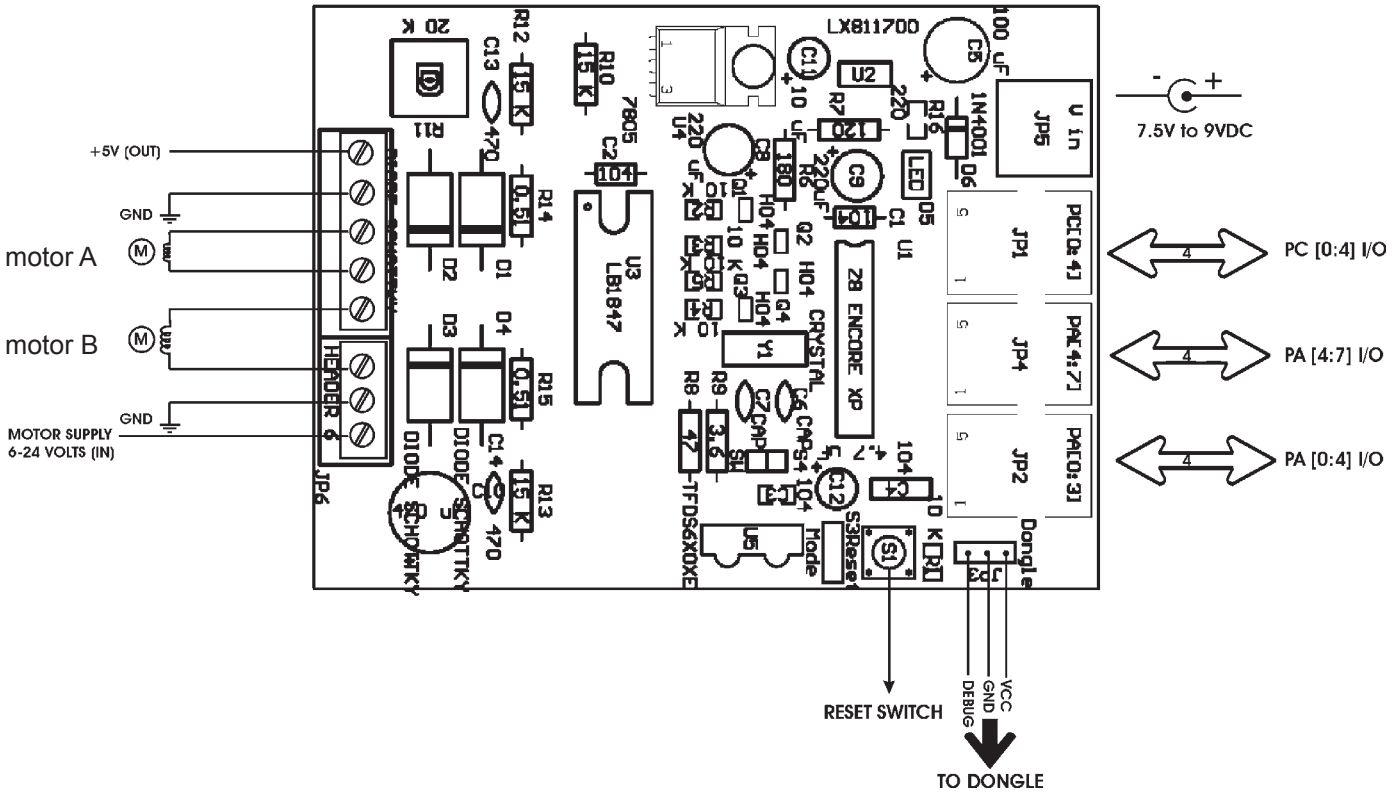


Figure 3. Wiring guide for the UPRC. The controller uses two separate supplies, one feeds the controller, and the other powers the motor system.

© 2007 e-Gizmo Mechatronix Central

Connecting a Debugging Cable

The debugging cable connects your UPRC with the PC's ZDS II development environment during the program writing and debugging process. The wafer side of the debugging cable connects to JP3 of the controller. Connect with the labeled side facing the outside edge of the PCB.

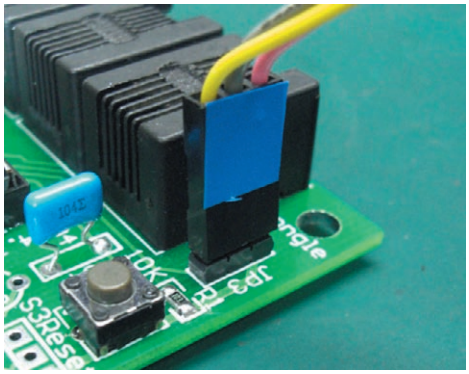


Figure 4. The debugging cable, or dongle, connected on JP3.

Installing an External Crystal

For most applications, the internal 5.53MHz internal precision oscillator is fast enough to do even complex task. If your application demands a faster processing speed, then you can install an external crystal with up to 20MHz frequency, as shown in the figure 5 and 6. A pair of loading capacitor is necessary to ensure sustained oscillation. Value of loading capacitor depends on the crystal; you can start with a pair 15pF capacitor if you are not sure where to start. You should remember that installing an external crystal prohibits you from using PA0 and PA1 (JP2 pin 2 and 3) for any other purpose, and absolutely nothing, not even a short dangling wire, should be left connected on these pins. Operation with external cystal should then be enabled through program control.

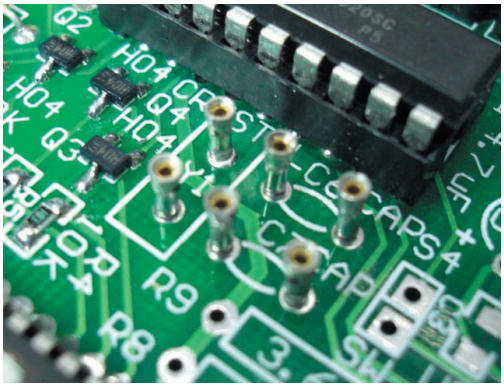


Figure 5. Pin sockets for external crystal.

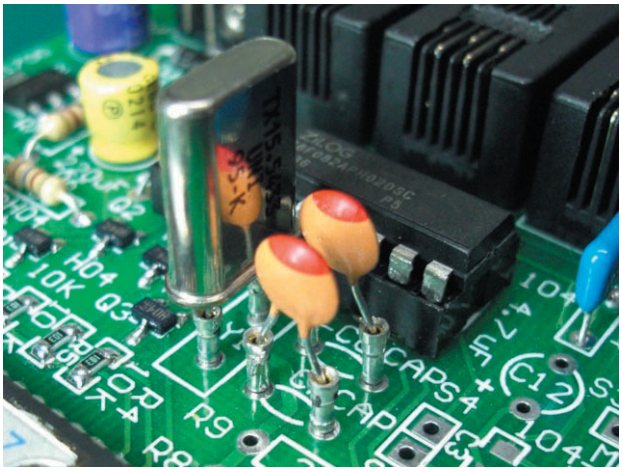


Figure 6. External crystal and loading capacitors can be added simply by plugging these components into their corresponding pin sockets.

Using the UPRC Motor Driver

The Motor Driver chip, LB1847, in reality is a PWM bipolar stepping motor driver chip. It has a dual H-bridge driver, controlling output current in PWM mode. The dual H-bridge driver can be controlled independently; that gave us the consent to use it as a dual DC motor driver.

Figure 3 shows the wiring diagram for a two DC motor application. Of course, you can omit the other motor if you do not need it. Use a Vmotor that is about 2V higher than the rated motor voltage to compensate for the voltage drop introduced by the LB1847. Adjust the current control trimmer R11 fully counterclockwise to ensure maximum output current drive.

The motors can then be controlled through PB0 to PB3 as shown in Table1 and Table 2.

Table 1 : Motor A control

PB0	PB1	Motor A
x	0	OFF
0	1	Forward
1	1	Reverse

Table 2: Motor B control

PB2	PB3	Motor B
x	0	OFF
0	1	Forward
1	1	Reverse

x- don't care

Figure 7 shows the motor circuit driving a stepping motor. The circuit works best with 4-wire bipolar stepping motor. A 6-wire unipolar stepping motor can be used as well and may be wired as shown in figure 8. You should adjust the R11 according to the rated current of the stepping motor. When R11 is rotated fully counter clockwise, current is at is maximum setting of 1.5A. Fully clockwise rotation results in near zero current. Current is roughly linear with the trimmer adjustment position. Halfway position gives approximately 1.5/2 = 0.75A PWM current drive.

The motor can then be run by sequencing PB0 to PB3 as shown in table 3:

Table 3 : Stepping Motor control

PB0	PB1	PB2	PB3	Motor
x	0	x	0	OFF
0	1	0	1	step1
1	1	0	1	step2
1	1	1	1	step3
0	1	1	1	step4

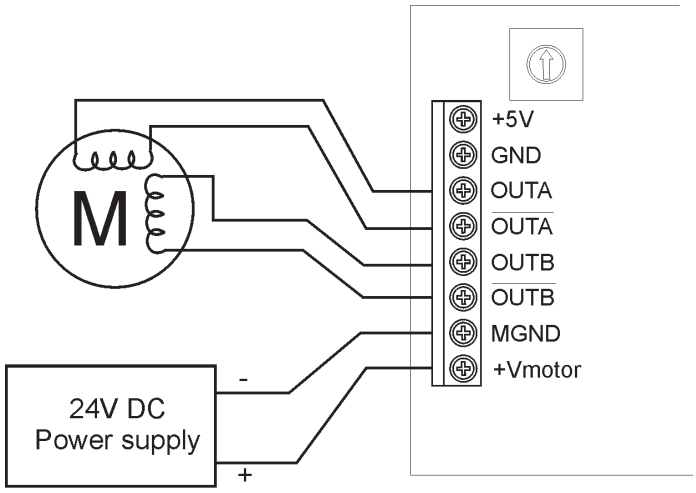


Figure 7. UPRC driving a 4-wire bipolar type stepping motor.

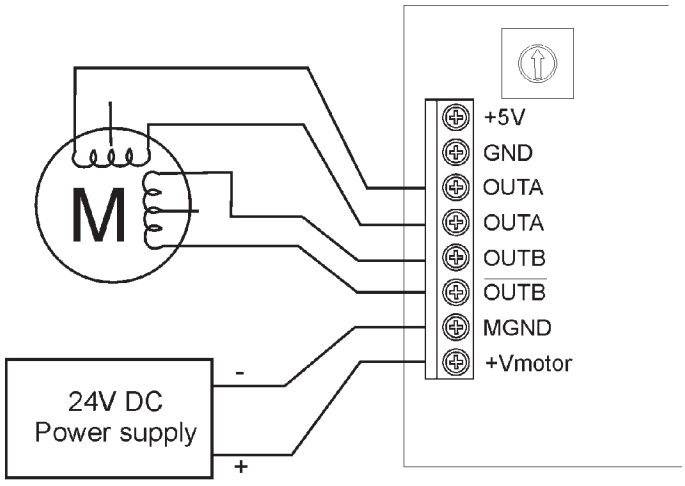


Figure 8. Even six-wire unipolar stepping motors will work with the motor driver circuit. Leave the center tap connections of each motor coil unconnected.

© 2007 e-Gizmo Mechatronix Central

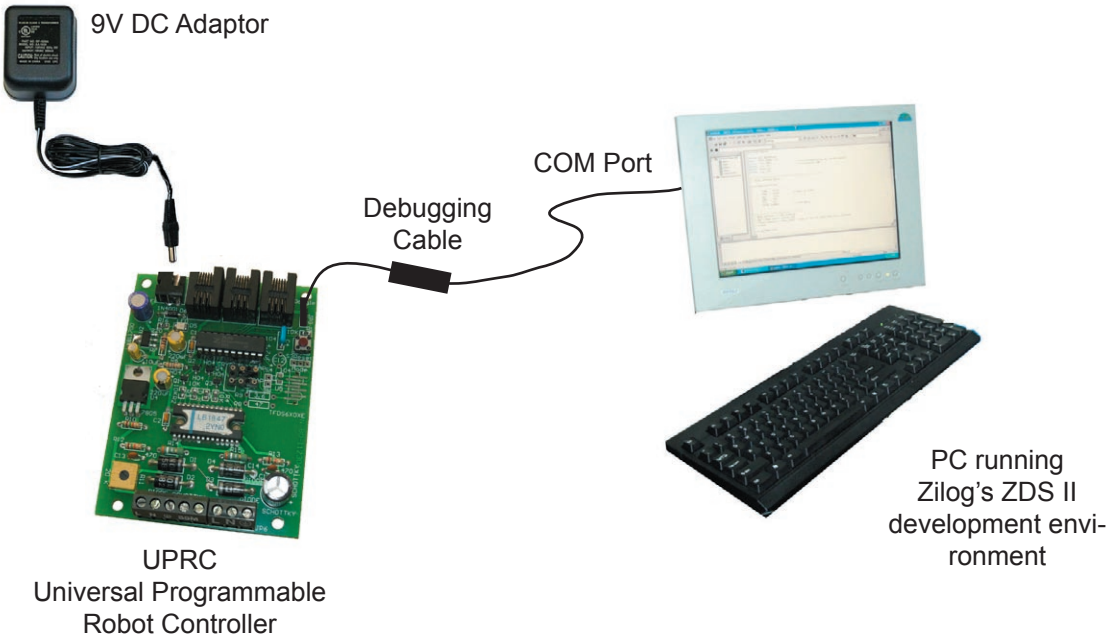


Figure 9. Programming and debugging setup. You do not have to remove any component from the UPRC system to program and debug it.

PROGRAMMING AND DEBUGGING SETUP

Equipment Required:

PC with Microsoft Windows XP, Windows 2000 SP4 or Windows 98 SE/ME. Zilog recommends a PC host running on a Pentium III or higher processor, with at least 128MB RAM. UPRC debugging cable requires one serial port (COM port), make sure your PC have it. You can use a USB to COM converter cable if you need to, just ready yourself to the extra fiddling that may be required to get your system to work with it.

Zilog's ZDS II development environment. This freeware package contains everything you need to program and debug the Encore! chip. This includes the editor, C compiler and assembler. You can download their latest version from www.zilog.com.

Debugging cable. This allows you to connect the UPRC to your PC during the programming and debugging session.

AC-DC adaptor to power your UPRC. Use a 9V adaptor with 250mA or higher output.

The programming and debugging setup is fairly simple. Just connect the debugging cable PC end to a PC COM port, and the other end to the UPRC module, and you can immediately start programming and debugging your Universal Programmable Robot Controller system.

APPLICATION IDEAS

There are many things you can do with the UPRC, bounded only by your imagination. The following pages illustrates some example applications you can work with your UPRC.

Note: Although the figures does not show the power supply connections of the UPRC, you are not smart if you omit these in your actual setup.

Interfacing with 5V Logic Circuits.

The abundance of TTL and 5V CMOS circuit create an assurrance that sooner or later, you will have to connect the UPRC into one.

Figure 10 illustrates one posiible setup. The A port PA2 to PA7 will accept 5V input safely, as long as these ports are configured in your program as inputs with no internal pull-ups. No external interfacing is necessary.

To drive a 5V TTL load, you may have to use a level converter. Fortunately, the XP device has a guaranted logic 1 output of 2.4V, which happens to be the same as LSTTL specifications. This tells us that we can omit the level converter, connect the UPRC output to a TTL input directly, and can get away with it (most of the time).

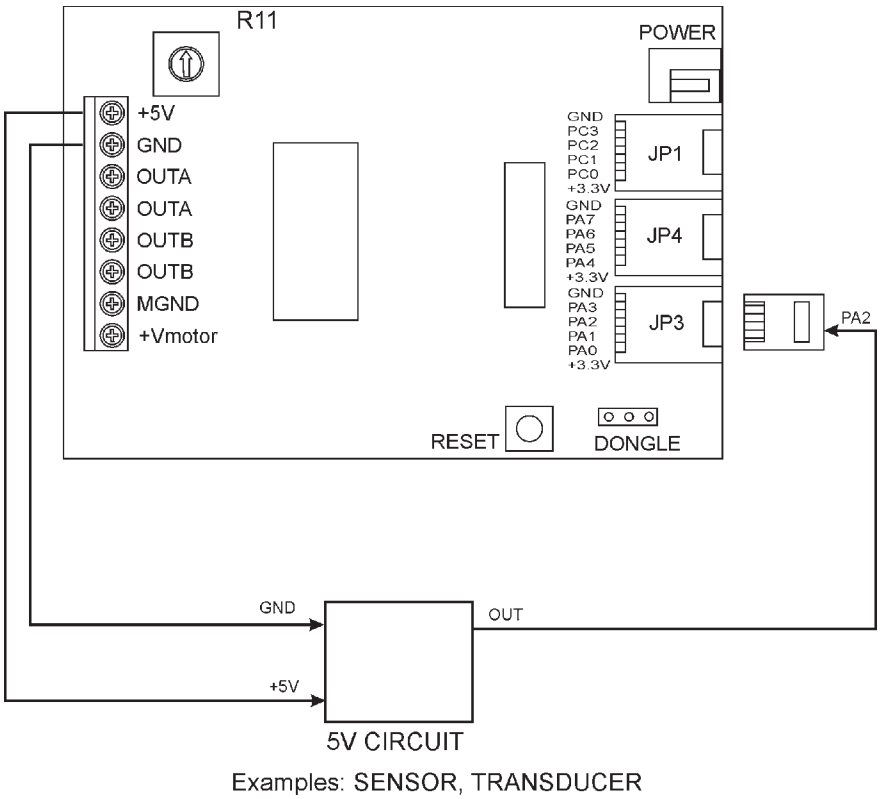


Figure 10. Connecting a 5V circuit with a UPRC. The circuit can be a sensor, transducer, or any other +5V logic circuit add-ons.

DC Servo Motor

Figure 11 shows one possible way to build a servo motor positioning controller using the UPRC. The position sensor, a potentiometer VR1 in this case, is mechanically linked to the motor. The motor movement and position is measured by examining the voltage across the wiper arm of the potentiometer. This is done by the UPRC through any one of its ADC port. In servo applications, the potentiometer voltage will tell the UPRC if the motor is up to the desired position. If not, the UPRC will then execute a sequence of programs until the desired motor position is reached. R and C filters out the noise that may appear with the voltage coming out of the potentiometer arm as it rotates. Note that the potentiometer has limited rotational life span, and this vary greatly from one manufacturer to another. You should always pick one with a specified and long rotational life span. Pick an R1 value that will limit the voltage across the potentiometer to ADC's Vref.

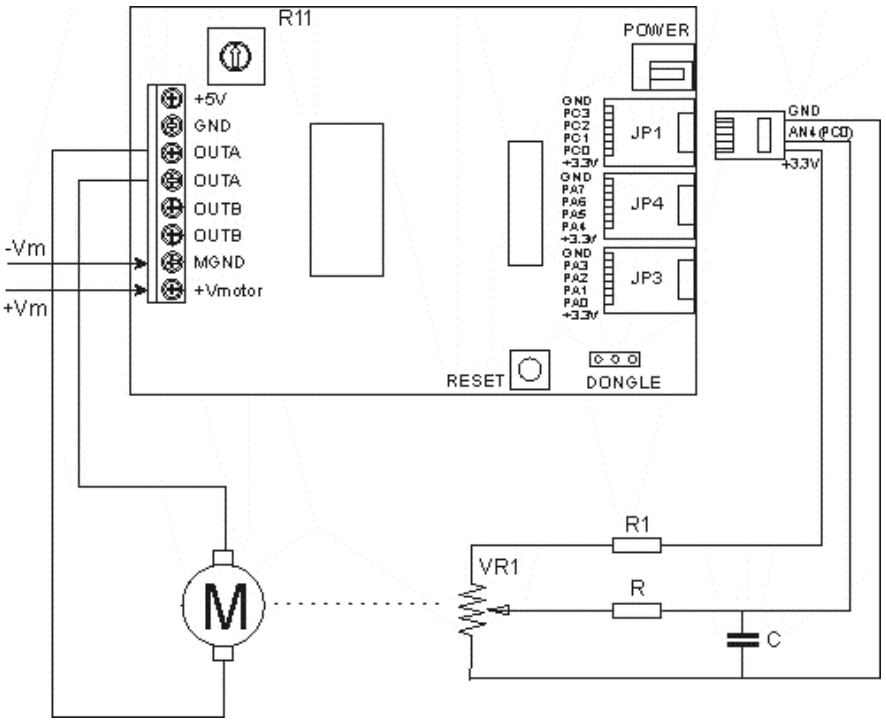


Figure 11. A motor positioning application (Servo Motor).

© 2007 e-Gizmo Mechatronix Central

Multiple UPRC in a System

You may encounter some applications requiring the use of two or more UPRC in a system. In multiple UPRC configuration, coordination between the UPRC is essential to ensure efficient and safe operation.

There are different ways to ensure an orderly operation with a multiple UPRC system. The easiest one to implement is to make one UPRC act as the master controller, the others as smart slave. Multiple masters is possible, and can even result to a more powerful and flexible system, but is generally harder to tame.

Whatever configuration you finally decide to take, multiple UPRC system will require a communication channel where commands and informations between UPRCs can be passed on. The UART channel fits this function well and can be implemented easily, as shown in figure 12. Data rates in excess of 128kbs is possible with short wire runs. Keep the wire lenght connecting the two UPRC as short as possible, not to exceed 3 feet end to end.

Note that this simple approach can only be implemented with two UPRC system. Three or more will require the use of RS-485 bus.

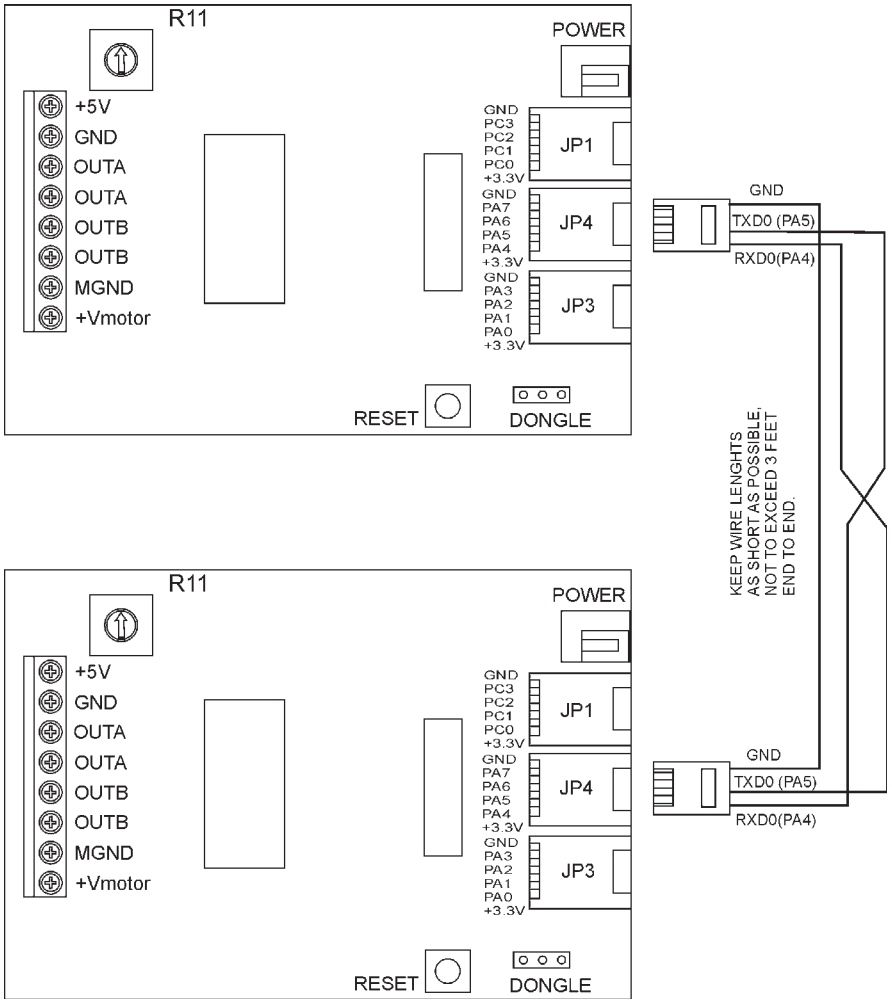


Figure 12. Using the UART channels to establish a communication link between two UPRCs.

Figure 13. Suggested UART wiring scheme. Three GND lines shields the signal lines Rx and Tx to keep the system resistant to externally introduced noise.

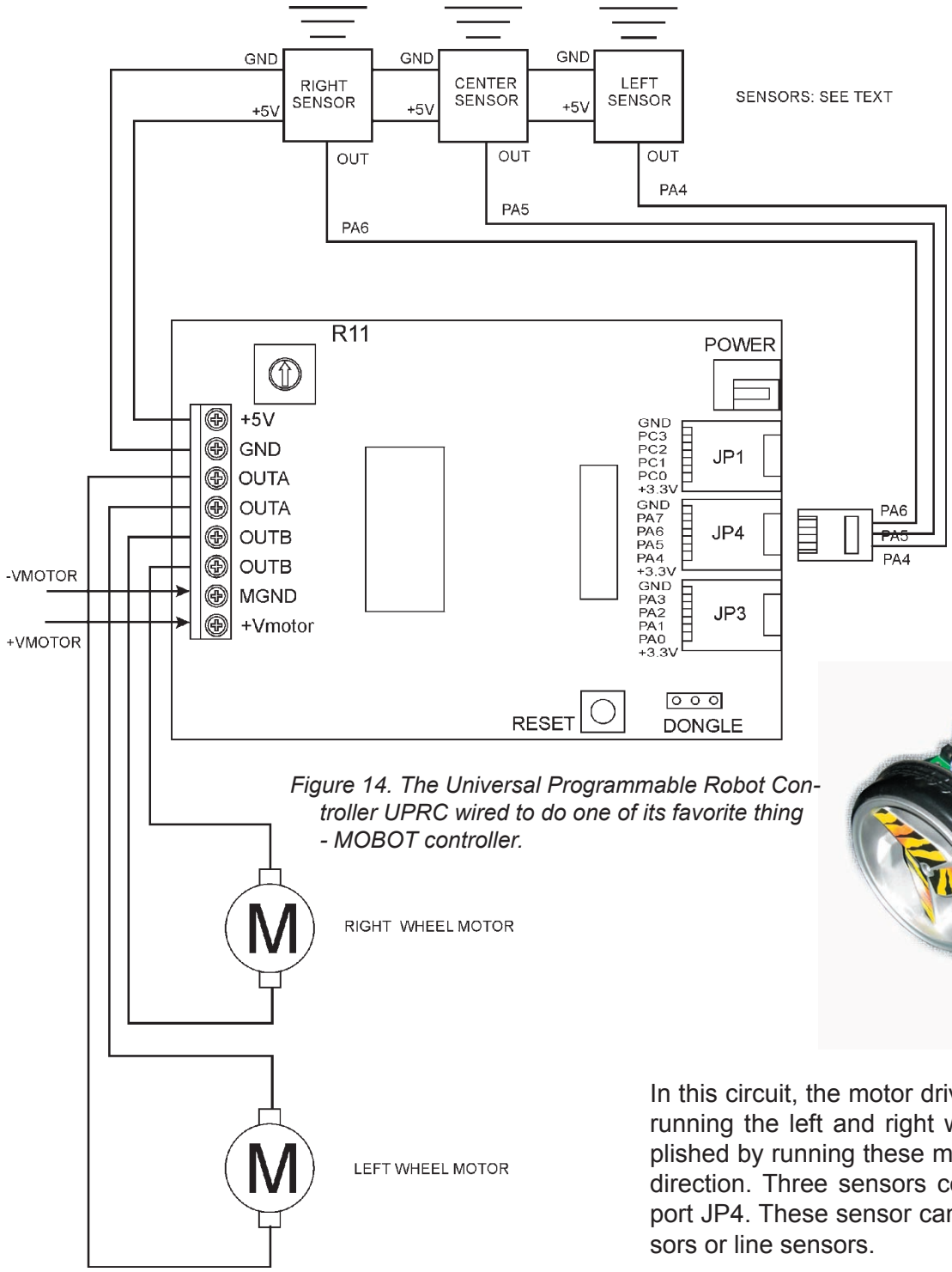


Figure 14. The Universal Programmable Robot Controller UPRC wired to do one of its favorite thing - MOBOT controller.

MOBOT APPLICATION CIRCUIT

The UPRC was built with this application in mind. Generally considered as entry level robot, building the Mobile Robot, or MOBOT, is an excellent way to learn the principles of robotics hands on. This example will let you build and program a mobot that can follow a line path, or navigate through a maze of objects while avoiding collision.

In this circuit, the motor driver control two DC motors running the left and right wheel. Steering is accomplished by running these motors at different speed or direction. Three sensors connects to one set of I/O port JP4. These sensor can be infrared collision sensors or line sensors.

The collision sensor is a normally high switch, it will switch to logic 0 if an object blocks its path. The three sensor arrangement will let the UPRC detect obstacle up front, to the right, and to the left.

The line sensors switch to logic 1 if positioned over black lines. The three sensors will tell UPRC if it is on path, a little off to the left, or a little off to the right.

An example source code that demonstrates these functions can be downloaded at UPRC page of www.e-gizmo.com.◙

© 2007 e-Gizmo Mechatronix Central