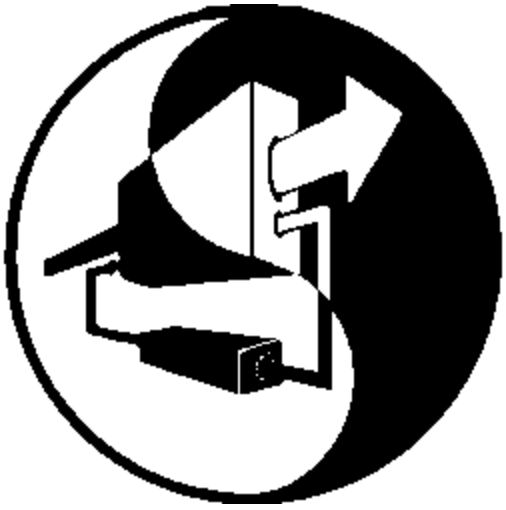


CY545

High Performance
Stepper System Controller

Copyright and Disclaimer Notice

Cybernetic Micro Systems, Inc. software products are copyrighted by and shall remain the property of Cybernetic Micro Systems, Inc. Duplication is subject to a license from Cybernetics. Cybernetic Micro Systems, Inc. reserves the right to improve design or performance characteristics. Cybernetic Micro Systems, Inc. assumes no responsibility for the use of any circuitry other than circuitry embodied in Cybernetic products. No other circuit patent licenses are implied.

Information furnished by Cybernetic Micro Systems, Inc. is believed to be accurate and reliable. However, no responsibility is assumed by Cybernetic Micro Systems, Inc. for its use, nor for any infringements of patents or other rights of third parties which may result from its use. No license is granted by implication or otherwise under any patent or patent right of Cybernetic Micro Systems, Inc. Further, Cybernetic Micro Systems, Inc. reserves the right to revise this publication and to make changes from time to time in the content hereof without obligation to notify any person or organization of such revision or changes; and Cybernetics assumes no responsibility for any errors which may appear in this document and makes no commitment to update the information contained herein.

Because of the wide variety of applications in which this product may be used, Cybernetic Micro Systems makes no claim as to the product's fitness for any given application. It is therefore the user's responsibility to ensure the safety and viability of using this product in his application. This product is not recommended for use in critical life sustaining applications.

The following are trademarks of Cybernetic Micro Systems, Inc:

Bin-ASCII	CY250	CY512
CYMPL	CY300	CY525
Analog-ASCII	CY325	CY545
ASCII-Analog	CY327	CY550
CY123	CY360	CY600
CY132	CY480	CY750
CY232	CY233-LINC	P-51
CY233	CY500	USB-RAM

All rights reserved; no part of this publication may be reproduced, except for personal and archival uses, without the prior written permission of:

Cybernetic Micro Systems, Inc.
P.O. Box 3000, San Gregorio, California 94074 USA.
Tel: 650-726-3000 • Fax: 650-726-3003
www.ControlChips.com • Info@ControlChips.com

01/28/02

TABLE OF CONTENTS

Table of Contents

Table of Contents ..... **ii**
Table of Contents ii

Chapter 1 - Introduction to the CY545 ..... **1**
CY545 Stepper System Controller 1
Basic Features 1
Motion Control Functions 2
Motor Support Signals 2
General Command Functions and Special Features 3
Prototyping Board Support 3
CY545 Stepper Motor System Controller 4

Chapter 2 - Pinout & Package Overview **5**
CY545 Dip Pinout & Dimensions 5
CY545/J PLCC Pinout & Dimensions 6
CY545 DIP & CY545/J PLCC Pin Descriptions 7

Chapter 3 - Command Interfaces **11**
Command Interfaces 11
Parallel Interface 11
CY545 Interface to CY233 Network Control Chip 11
The CY545 Serial Interface 12

Chapter 4 - Command Summary **15**
CY545 Command Summary 15
Command Formats 16
ASCII Command Format 16
Binary Command Format 17

Chapter 5 - Motor Control Commands **18**
Motor Control Commands 18
Set Stepping Rate 18
Set Count and Direction 18
Set Step Mode 19
Moving the Motor 20
Position Counter “Wrap-around” 20

Chapter 6 - Bit Function Commands **21**
Bit Function Commands 21
User Bits 21
Bit Dependent Jump and Wait 22
Step/Test/Seek Home Command 24
Special Note Regarding Older Versions 24

Chapter 7 - External Memory Commands 25
External Memory Support Commands 25
External Memory Address Pointer 25
Writing Commands into Memory 25
Running From External Memory 26
Example 28

Chapter 8 - Program Branch Commands 29
Jump and Loop 29
Nested Loops 29

Chapter 9 - Operating Mode Command 31
Operating Mode Command 31

Chapter 10 - Miscellaneous Commands 35
Initialize Command 35
Time Delay Command 35
Query Command 36
Display CY545 Status 36
Display Memory Contents 37
Message Display Command 38
Embedded Control Characteristics and Their Use: 38
Hewlett Packard (HP) LED 8 Digit Display 38
LED Display Details 39

Chapter 11 - CY545 Example Circuits 40
Three Example Cases 40
CY545 Example Circuit – Minimum Serial Configuration 40
CY545 Example Circuit – Minimum Parallel Configuration 41
CY545 Example Circuit – Maximum Configuration 42

Chapter 12 - External Memory Support 43
Memory Read/Write 43
“Auto-Start” Feature for Stand Alone Operation 44

Chapter 13 - Thumbwheel Switch Support 46
Thumbwheel Switch Support 46
CY545 Thumbwheel Switch Parameter Addresses 49
CY545 Thumbwheel Circuit Diagram 50

Chapter 14 - CY545 Output Display Support 51
Output Format 51
Output to LCD (CY325-Based) 51
Output to Parallel Device (Including CY233 Network Chip) 52
Output to 8-Digit LED Display 52

Chapter 15 - CY545 Prototyping Support 54
The CYB-545 Prototyping Board 54
CYS-545 Panel Software 55
CyberCom Console Software 55
The CYB-233 Prototyping Board 56
The CYB-003 Prototyping Board 57

Chapter 16 - Timing and Control....58
Timing and Control....58
The Parallel Handshake....58
Immediate Commands During External Memory Program Execution....59
The Serial Interface....62
External Memory Control Signals....63
Special HP LED Display Control Signals....65
Thumbwheel Switch Interface Control Signals....66
Stepper Motor Interface Signals....67
PULSE and CWW....67
STOPPED....69
SLEW....69
INHIBIT_ABORT....70
CW_LIMIT and CCW_LIMIT....71
JOG Mode Operation....72
Automatic Position Display....73
Home Signal....73

Chapter 17 - CY545 Step Rate Information....75
CY545 Step Rate Information....75
The Slope Parameter....75
Optimal Acceleration Curves....76
Acceleration Curve as a Function of R....76
Slope and Elapsed Time....77
CY545 Slope Curves....78
CY545 Step Rate Table @ 11 MHz....81
CY545 Step Rate Table @ 12 MHz....82
CY545 Step Rate Table @ 14.75 MHz....83
CY545 Step Rate Table @ 16 MHz....84

Chapter 18 - Electrical Specifications....85
CY545B Electrical Specifications....85
Electrical Conventions....85
Reset Circuitry....85
Clock Circuits....86

Chapter 19 - Circuits and Examples....87
Variable Stock Cut and Punch....87
Driver Circuit Considerations....89
Interfacing to the 8255....92
Operating Several CY545s Using a Common Data Bus....93
Synchronization of Two CY545s....93
Coordination of Several CY545s....94
Example Programs and Waveforms....95
ASCII and Binary Data Programming Examples....96

Chapter 20 - Getting Your CY545 Running....97
Getting Your CY545 Running....97
BASIC Language CRT Emulation Program....99
Changes from CY545 to CY545A....101
Changes from CY545A to CY545B....101
Stepper Motor Controller Selection Guide....102

CHAPTER 1 - INTRODUCTION TO THE CY545

CY545 Stepper System Controller

In 1979 Cybernetic Micro Systems invented the first Single Chip Stepper Motor Controller, the **CY500**. Since then we have continued to design improved versions, the **CY512** (1981), the **CY525** (1983). This manual introduces our (1988) **fourth generation device, the CY545 Stepper System Controller**. A fifth generation device (1993) the **CY550** is also available. The CY545 is the first **Stepper Motor System Controller chip**, to offer completely new features, such as:

- 27,000 Steps/Sec
- Million step motions
- Serial or Parallel interface
- LED display interface
- LCD display interface
- Thumbwheel switch interface

We believe this device offers all of the features you want to implement sophisticated motion control systems at very low cost.

For example, our CY500, CY512 and CY525 controllers work best in systems with a host computer that issues single commands, or sequences of commands as a program, to the controller, and then tends to other system functions while the Stepper Motor Controller is running the motor. Maximum performance is gained when the host CPU is closely coupled to the Stepper Motor Controller IC.

While the CY545 is also capable of working with a host CPU in a closely coupled fashion, the CY545 contains extensions that allow it to function very well in a stand-alone system. In fact, it is possible to base a simple controller design around the CY545, and operate a machine with no CPU at all, bringing the benefits of intelligent, programmable stepper motor control to applications that do not justify a host computer system.

Basic Features

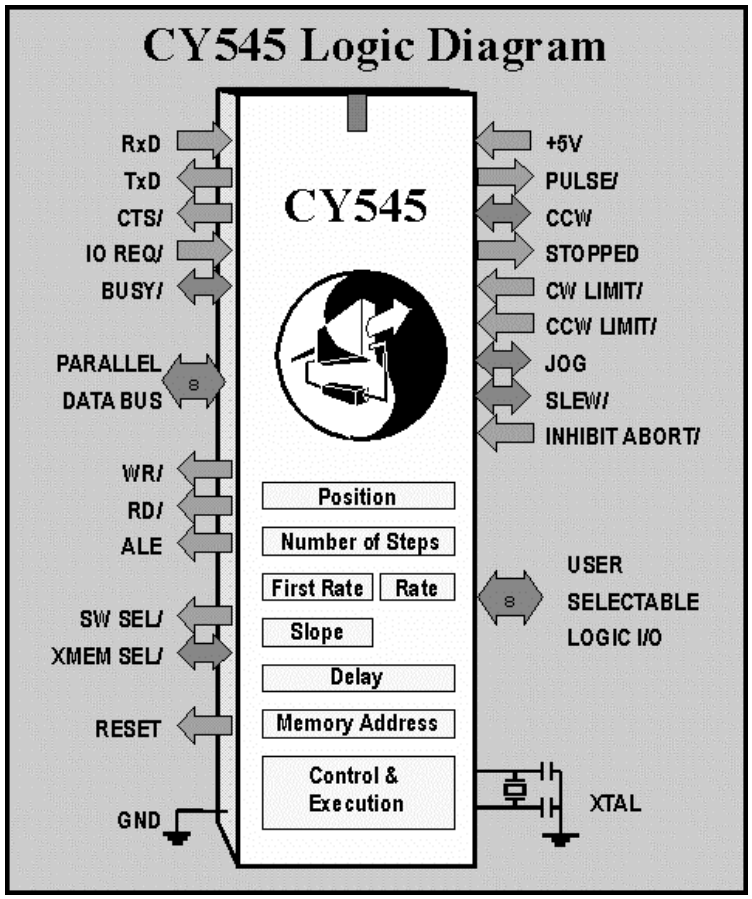
- Single 40 pin 5 volt CMOS integrated circuit
- Simple ASCII letter based commands and parameters
- Simple application for basic stepper motor functions
- Operation with a host computer or stand alone
- Built-in Display interfaces for alpha-numeric LED or LCD

Motion Control Functions

- Programmable step rates from 20 steps/sec to over 27,000 steps/sec provide extremely large dynamic range.
- Linear acceleration curves for optimum performance.
- Partial accelerations make short moves in the least time.
- Separate parameters for starting rate, final rate, and acceleration values allow totally programmable motions.
- Relative moves of +/- 16 Mega steps from current position. Absolute moves within a 16 Mega step range.
- Continuous moves with no specific number of steps, allow acceleration from the starting rate to the final rate, followed by an indefinite run length.
- Internal step position updated during all motions.

Motor Support Signals

- Step Pulse and Direction signals work with standard power driver modules.
- Internal or external direction control.
- Forced ramp down and abort signal for emergency or externally controlled end of motion.
- Separate CW and CCW limit signals inhibit stepping beyond one limit, but allow stepping in the opposite direction.
- Moving/not moving signal for use as Motion Complete or to switch stepper power driver between high and park power.
- Externally operated jog mode, with direction control and start/stop control from one signal, all at programmable manual control step rates between 1 and 1K steps/sec.
- Automatic home sensor seek, with backlash compensation.

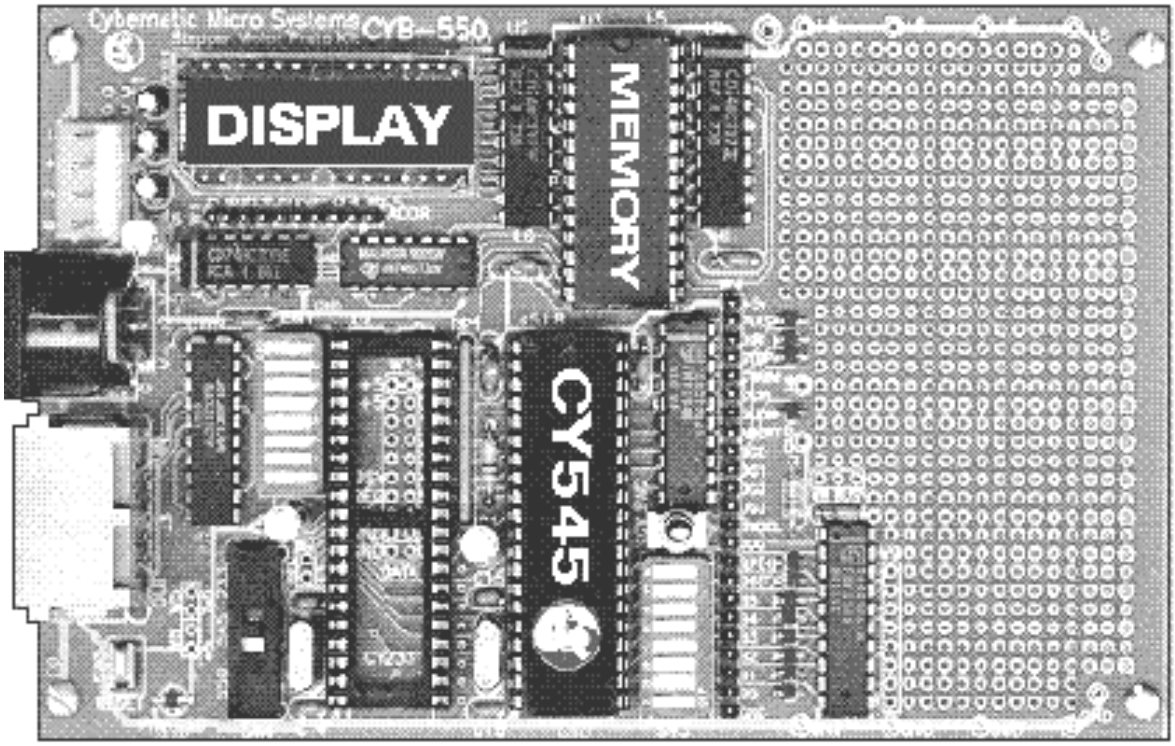


General Command Functions and Special Features

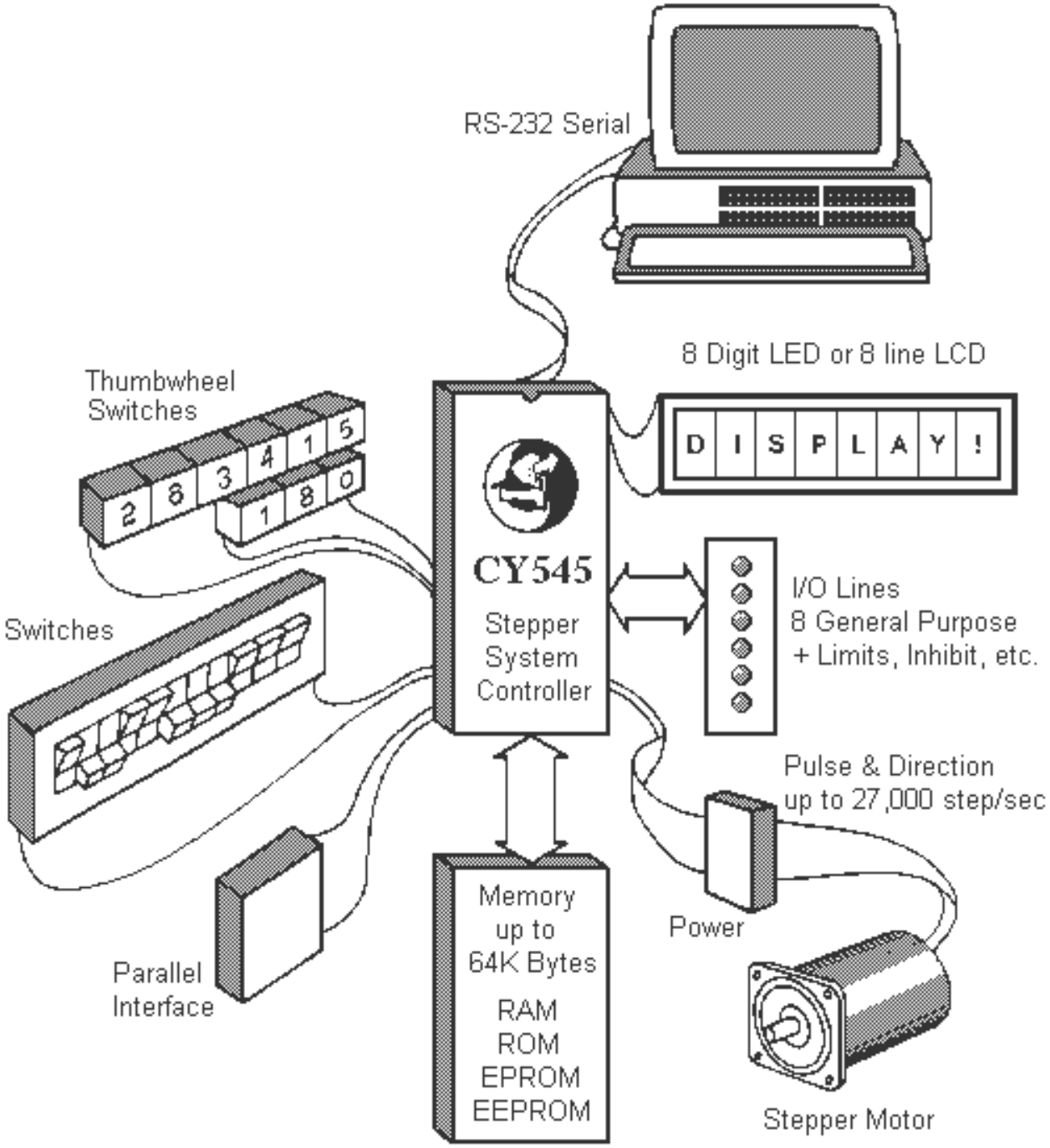
- Parallel command interface with two line handshake, compatible with CY233 network communications controller.
- Serial command interface with fixed or adaptive baud rate selection.
- Support for up to 64K of optional external command memory implemented as RAM, ROM, EPROM, or EEPROM.
- Support for optional external setting of parameter values, through standard thumbwheel switches.
- Selectable display for output messages from standard serial, standard parallel, or CY233 interfaces, plus special support for a parallel HP HDSP-211x LED display, or a CY325 based serial LCD display.
- User controlled, multi-purpose I/O lines, for:
    - set and clear bit functions
    - test bit and branch functions
    - wait for bit value functions
    - automatic home seek functions

Prototyping Board Support

The CYB-545 Prototyping board implements most of the available features of the CY545 Stepper System Controller. A more complete description may be found in the chapter on Prototyping Support.

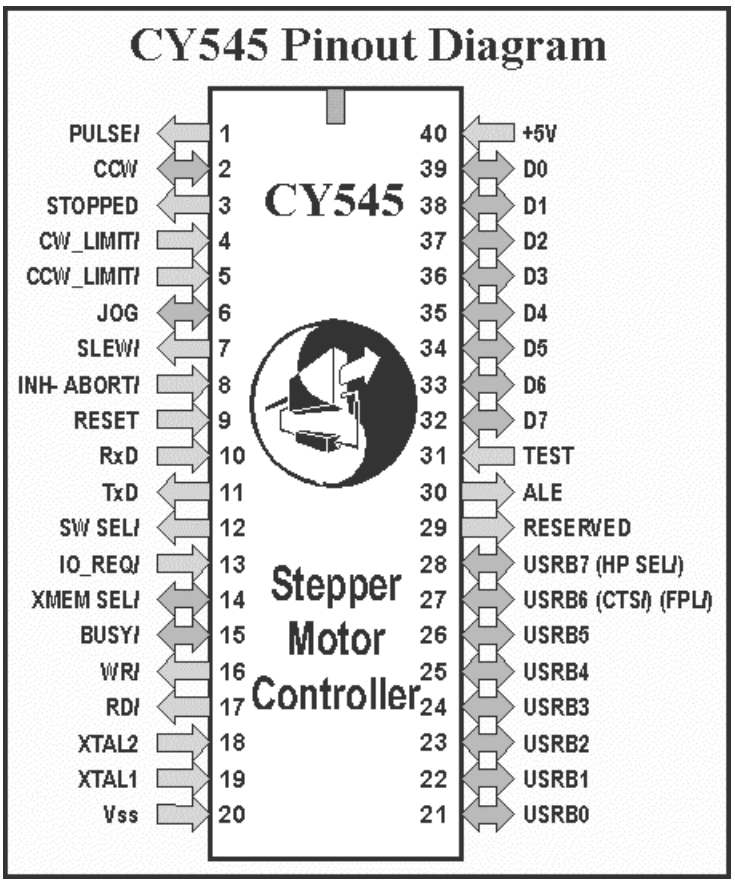


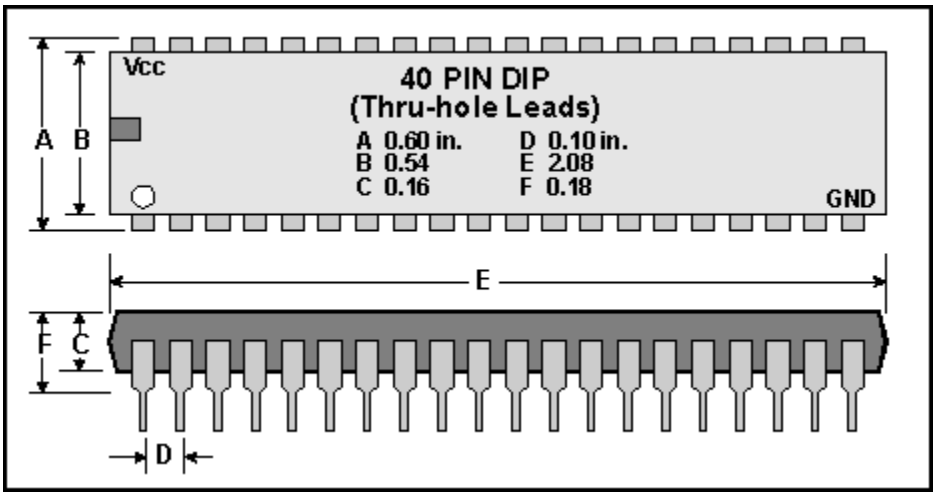
CY545 Stepper Motor System Controller



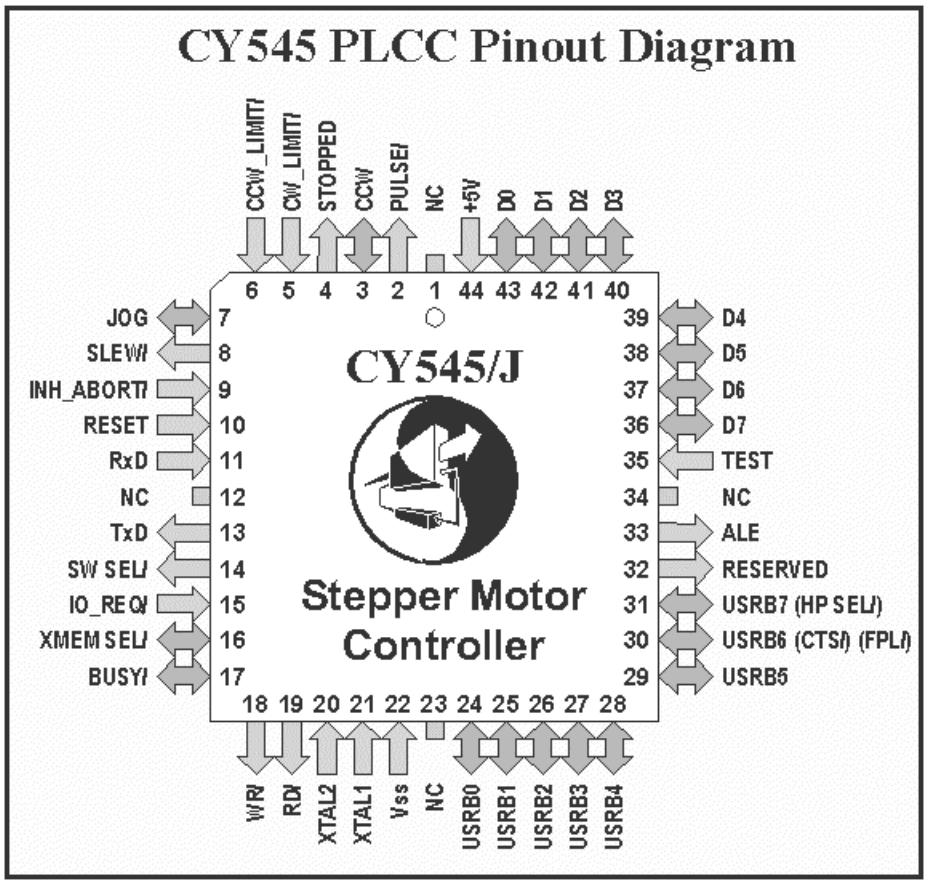
CHAPTER 2 - PINOUT & PACKAGE OVERVIEW

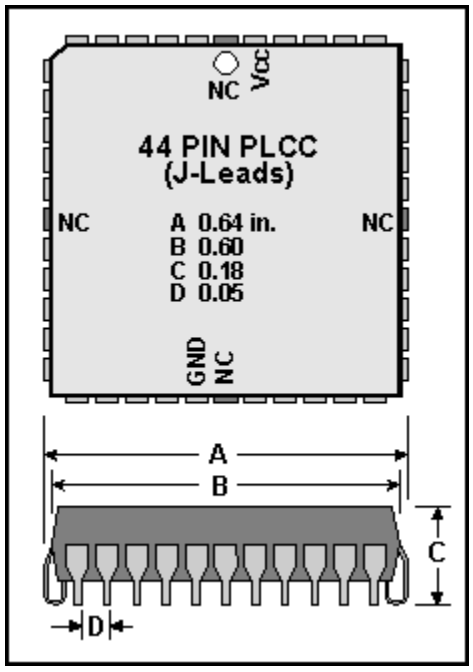
CY545 Dip Pinout & Dimensions





CY545/J PLCC Pinout & Dimensions





CY545 DIP & CY545/J PLCC Pin Descriptions

DIP	PLCC		Pin Name	Description
-	1	(O)	**NC**	No connection
1	2	(O)	**PULSE/**	Step pulse output, one pulse per step
2	3	(I/O)	**CCW**	Step direction

These pins run the stepper power driver. A high level on the direction signal indicates counter clockwise direction, while a low level indicates clockwise direction. The Pulse signal is normally high, going low at the beginning of each step.

DIP	PLCC		Pin Name	Description
3	4	(O)	**STOPPED**	Motion status, low while stepping

This signal may be used to indicate motion complete, by its high level. It may also control power selection for the stepper driver, switching between high power for stepping, and parking power while stopped.

DIP	PLCC		Pin Name	Description
4	5	(I)	**CW_LIMIT/**	Clockwise step limit reached
5	6	(I)	**CCW_LIMIT/**	Counter Clockwise step limit reached

If either signal is low, it indicates a limit has been reached. The CY545 will not continue stepping in that direction until the limit signal is removed. Normal stepping is allowed in the opposite direction.

DIP	PLCC		Pin Name	Description
6	7	(I/O)	**JOG**	Manual stepping control

This signal is the manual stepping control input. When it is floating, the CY545 will not jog. However, if the signal is connected to ground, the CY545 will jog in the CW direction. The jog step rate is derived from the First Rate parameter and the normal Step Rate Table, by dividing the selected rate by 20. This gives a range of about 1 step per second to about 1000 steps per second. The input signal is always enabled, and is tested by the CY545 while it is not executing another command.

DIP	PLCC		Pin Name	Description
7	8	(I/O)	**SLEW/**	Slew indicator signal
8	9	(I)	**INHIBIT_ABORT/**	External motion control

This signal group indicates and controls motion status. SLEW/ indicates when the maximum selected step rate has been achieved. If externally driven low at the beginning of a motion, it also selects the continuous step mode. INHIBIT_ABORT/ can hold off motion at the start or force an early down ramp, then stop motion when the CY545 has ramped down to the starting rate.

DIP	PLCC		Pin Name	Description
9	10	(I)	**RESET**	CY545 hardware reset, active high pulse
10	11	(I)	**RxD**	Received serial data into CY545
-	12	(O)	**NC**	No connection
11	13	(O)	**TxD**	Transmitted serial data from CY545

These are TTL level serial data signals that can be used to issue commands to or get replies from the CY545. Addition of an RS-232 voltage level driver/receiver, such as the Maxim MAX233, allows any RS-232 device, such as the IBM PC COM1 port to talk directly to the CY545.

DIP	PLCC		Pin Name	Description
12	14	(O)	**SW_SEL/**	Switch select

This signal is active when the CY545 is reading a parameter value from external thumbwheel switches. The switch address is generated by the parallel data bus.

DIP	PLCC		Pin Name	Description
13	15	(I)	**IO_REQUEST/**	Parallel handshake input signal

This signal is driven low when a parallel command is issued to the CY545. It is used as part of the parallel command handshake. If the parallel command function is not used, this signal may select a fixed serial baud rate as follows:
0 = 300 baud, 1 = 2400 baud, F = 9600 baud

DIP	PLCC		Pin Name	Description
14	16	(I/O)	**XMEM_SEL/**	External local memory select

This signal goes low when the CY545 reads or writes to the local external memory. Data and addresses are generated by the parallel data bus. Up to 64K bytes of external memory are supported. When this signal is driven low at power-up or reset, the CY545 will not test for the Auto Start key sequence or automatically execute commands from the external memory.

DIP	PLCC		Pin Name	Description
15	17	(I/O)	**BUSY/**	Parallel handshake acknowledgement

BUSY/ is used with IO_REQUEST/ to implement the two line parallel command handshake. This signal indicates that the CY545 has accepted a parallel data character from the device driving the IO_REQUEST signal. If the parallel handshake is not used, this pin may be tied low to select fixed baud rates through the IO_REQUEST signal.

DIP	PLCC		Pin Name	Description
16	18	(O)	**WR/**	Write strobe
17	19	(O)	**RD/**	Read strobe

These strobes are active while the CY545 is reading from the external thumbwheel switches, display, or memory, or while it is writing to the external display or memory.

DIP	PLCC		Pin Name	Description
18	20	(I)	**XTAL2**	Crystal connection
19	21	(I)	**XTAL1**	Crystal connection

An external crystal or clock source is connected to these pins, with a value between 3.5 MHz and 16 MHz. For serial communications, an 11 MHz crystal will give standard baud rates.

DIP	PLCC		Pin Name	Description
20	22	(I)	**VSS**	Power supply ground.
-	23	(O)	**NC**	No connection

DIP	PLCC		Pin Name	Description
21	24	(I/O)	**USRB0**	User selectable function, bit 0
22	25	(I/O)	**USRB1**	User selectable function, bit 1
23	26	(I/O)	**USRB2**	User selectable function, bit 2
24	27	(I/O)	**USRB3**	User selectable function, bit 3
25	28	(I/O)	**USRB4**	User selectable function, bit 4
26	29	(I/O)	**USRB5**	User selectable function, bit 5
27	30	(I/O)	**USRB6**	User selectable function, bit 6 or FPL/ or CTS/
28	31	(I/O)	**USRB7**	User selectable function, bit 7 or HP_SEL/

This group of signals is used for multiple, user selectable functions, including bit set and clear, bit test and branch, wait for signal value, and auto home functions. Any function may be performed on any bit. Bits 6 and 7 also have fixed alternative functions of CY233 FPL control and local parallel HP display selection if these features are used with the CY545. If the application does not use a CY233, CTS/, or HP display, these bits may be used for any other functions.

DIP	PLCC		Pin Name	Description
29	32	(O)	**RESERVED**	Reserved signal, not used by CY545
30	33	(O)	**ALE**	Address latch enable

The ALE signal is used to demultiplex the lower byte address from the data bus during external memory, display, or thumbwheel switch operations.

DIP	PLCC		Pin Name	Description
-	34	(O)	**NC**	No connection
31	35	(I)	**TEST**	Internal test signal, connect to Vcc.

DIP	PLCC		Pin Name	Description
32	36	(I/O)	**D7**	Parallel data bus, bit 7, MSB
33	37	(I/O)	**D6**	Parallel data bus, bit 6
34	38	(I/O)	**D5**	Parallel data bus, bit 5
35	39	(I/O)	**D4**	Parallel data bus, bit 4
36	40	(I/O)	**D3**	Parallel data bus, bit 3
37	41	(I/O)	**D2**	Parallel data bus, bit 2
38	42	(I/O)	**D1**	Parallel data bus, bit 1
39	43	(I/O)	**D0**	Parallel data bus, bit 0, LSB

This group is the parallel data bus, used to read or write parallel data. During access to external thumbwheel switches, memory, or display, the lower byte of the device address is presented on these lines just prior to the data transfer. Parallel commands are also accepted by the CY545 on these lines, using the handshake control signals. These lines are open drain, so external pull-up resistors are required when they are used as outputs.

DIP	PLCC		Pin Name	Description
40	44	(I)	**VCC**	+5 Volt power supply input.

CHAPTER 3 - COMMAND INTERFACES

Command Interfaces

The CY545 supports two basic command interfaces, a parallel interface and a serial interface. These signals are similar to functions on other CYxxx controllers from Cybernetic Micro Systems.

Parallel Interface

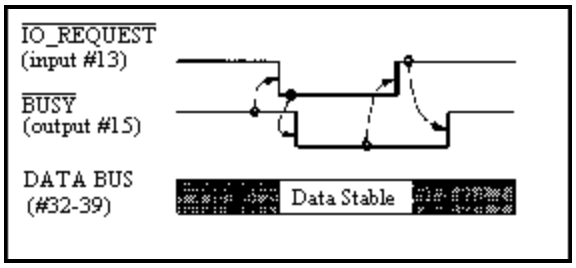
The parallel interface uses two handshake lines, IO_REQUEST and BUSY. When you wish to send a command character to the CY545, you first check that it is not busy; the BUSY signal should be high.

You then place the character to send on the parallel data bus lines. Be sure to drive all 8 data lines.

Next, you drive the IO_REQUEST signal low, indicating that a character is available

Now, wait for the CY545 to go busy, with a low level on the BUSY signal, indicating that the CY545 has read the command character.

Finally, you remove the character from the data bus and drive the IO_REQUEST line high again.

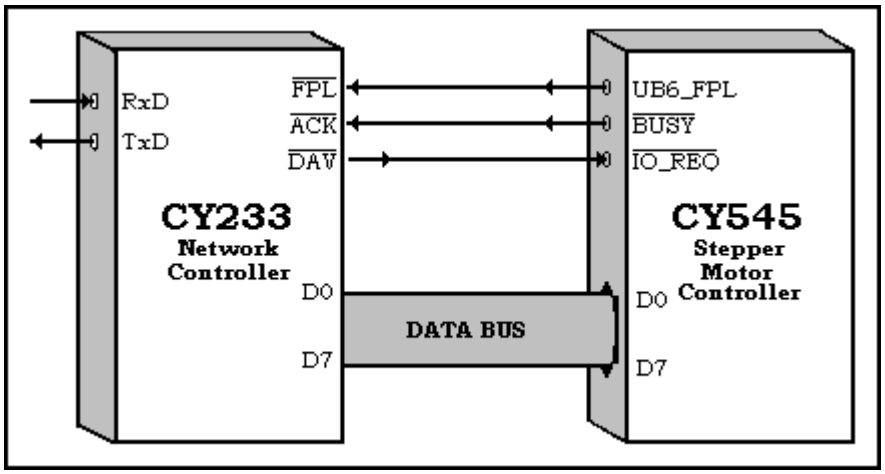


This function is repeated until all command characters have been issued to the CY545. In direct command mode, the CY545 executes every command as it is received. This means that the CY545 will go busy for a longer period of time with the last character of a command, while the command is actually executed. For long time delays or long motions, this time could be many seconds.

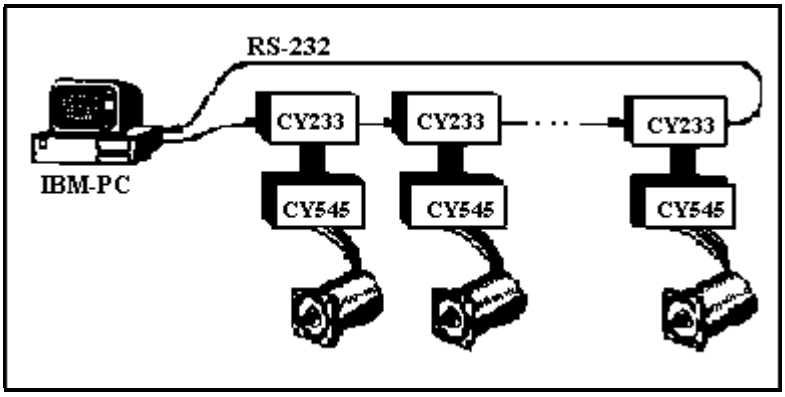
This parallel interface is compatible to that of other CYxxx control chips. It is also compatible with the CY233 Network Controller, so you can command several CY545s from a computer serial port, using a CY233 as a network “front end” to each CY545.

CY545 Interface to CY233 Network Control Chip

For a CY233-based interface, the DAV signal from the CY233 is connected to the IO_REQUEST signal of the CY545, and the ACK signal of the CY233 is connected to the BUSY signal of the CY545. In addition, the FPL signal of the CY233 is connected to USRB6 of the CY545, which shares the FPL function on this line. An example applications schematic, provided with this manual, shows the connections. The relevant portion is reproduced below.



The CY233 Network Control chip allows you to connect up to 255 devices to a single RS-232 serial communications line such as an IBM-PC COM1 or COM2 port, with unique addresses assigned to each device. The CY233 Network is ideal for distributed systems where central control is required but very high speed communication is unnecessary.



The CY545 Serial Interface

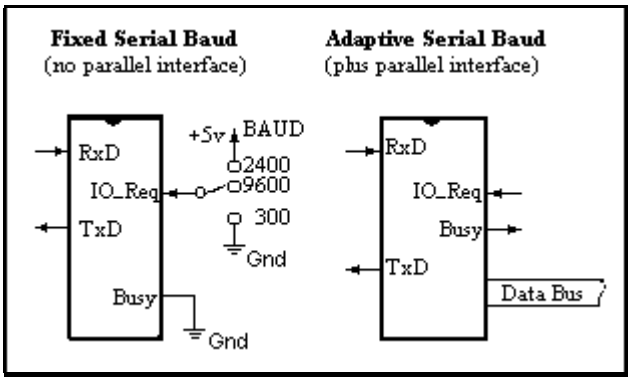
The CY545 also provides a direct serial command interface, which may be connected to a host computer or terminal. Since the CY545 signals are all TTL voltage levels, external RS-232 line drivers and receivers must be provided, to translate the RS-232 voltage levels to the CY545 TTL levels. A Maxim MAX-233 does this function in a single chip, operating at +5 volts. This interface is also shown on the applications schematic.

The serial interface may be operated in one of two ways, with a fixed baud rate, selected at power up, or with an adaptive baud rate, selected by two carriage return characters from the host.

The fixed baud rate mode is selected by tying the BUSY line low, so the CY545 will read it as a zero value on power up. The CY545 IO_REQUEST line value will determine the baud rate as follows:

F = 9600 baud
1 = 2400 baud
0 = 300 baud

When the IO_REQUEST line is left floating, 9600 baud is selected. When tied high, 2400 baud is selected, and when tied low, 300 baud is selected. For these fixed rates to be correct, the CY545 must be operated with an 11 MHz crystal clock. Other values will scale the baud rate linearly by the difference in crystal frequencies.



An alternative method of selecting the baud rate is through a parallel Mode command. For this command, nine baud rate selections are available, from 300 to 57600, plus the self adaptive rate. Details of the Mode command are described in a later section.

Note that the selection of fixed baud rate mode through the BUSY and IO_REQUEST lines, gives up the use of the parallel handshake interface, since the lines are used to select the baud rate. This is fine if only the serial interface will be used. However, if both interfaces must be used on the same CY545, the baud rate may only be set adaptively, or through a parallel Mode command.

The adaptive serial mode is chosen by default, when the BUSY signal is left floating, so the CY545 can drive it at power up. In this mode, the serial baud rate is not set until the CY545 receives two carriage return codes. Be sure to send these characters after power up or any reset (hardware or software). Once the two carriage returns have been received, normal CY545 commands may be sent.

When working with the serial interface, the best crystal frequency for the CY545 is 11 MHz. This will provide the largest baud rate operating range, with the least baud rate error. Internal timer resolution of the CY545 will limit the available rates, especially at the high end. At 11 MHz, standard rates up to 19200 baud may be selected. Note that other CY545 functions, such as the step rates and time delays, have been calibrated for a 12 MHz clock source. This means these functions will run somewhat slower when 11 MHz is used. Changes to the relevant parameter values can compensate for some of this difference. Alternatively, a 12 MHz clock will work fine, if the selected adaptive baud rates are always 4800 baud or less.

When operating the CY545 at 16 MHz, internal timer resolution still limits the reliable operation of the serial interface to 4800 baud or less. The error at higher baud rates may be too great for consistent communications.

However, there is a crystal frequency of 14.7456 MHz at which the CY545 can be operated in the adaptive mode. With this frequency, the internal resolution of the CY545 baud rate timer allows it to adapt to all standard baud rates from 300 to 19,200 baud, plus two higher, non-standard rates of 38,400 and 76,800 baud.

Also, the CY545 serial interface works on a fixed format character, which is always 8 data bits, with no parity bit, and one stop bit. All 8 data bits are used to interpret the command character values, so be sure to send the proper 8 bit codes for the various ASCII command characters.

Finally, the CY545 optionally implements a Clear to Send signal ("CTS/") which may be used by the host system to indicate when the CY545 is busy processing a command. No additional characters should be sent to the CY545 while CTS/ is off (TTL high). Once the CY545 has finished processing the current command, CTS/ will be enabled again, indicating that the CY545 is ready for more serial characters. The polarity of the CY545 CTS/ signal allows it to be connected directly to your host system CTS RS-232 signal, through a standard RS-232 line driver. No additional logic is required.

The CTS/ signal shares functions with User Bit 6, which is also the CY233 FPL/ signal. Since the signal has multiple functions, the CTS/ function is disabled at power on or any restart operation. You must issue a Mode command with MBit 5 set in order to enable the CTS/ signal. See the later section on the Mode Command for more details.

CHAPTER 4 - COMMAND SUMMARY

CY545 Command Summary

The following is a list of CY545 commands, including the command letter and argument structure. Arguments without a suffix should be single byte numeric values up to 255, while arguments with a "16" suffix may be two byte numbers up to 65535, and arguments with a "24" suffix may be three byte numbers up to 16777215. Commands are entered as the single upper case ASCII letter, followed by a space, and the argument value. Multiple arguments may be separated by a space or comma. The command ends with a carriage return character. Commands without arguments simply use the command letter, followed directly by the carriage return.

Command	Function
A Pos24	**A**t position, sets current step position.
B Bit	**B**it set or clear of user selectable bits.
C	Set **C**ontinuous step mode.
D Del16	**D**elay for specified milliseconds.
E	**E**nter following commands to external memory.
F Rate	Specify **F**irst or starting step rate.
G	**G**o step. Starts relative mode stepping.
H Bit	Seek **H**ome, using specified bit.
I	**I**nitialize controller, perform software reset.
J Addr	**J**ump to byte address of current memory page.
L Cnt,Addr	**L**oop to byte address for specific count.
N Num24	Set **N**umber of steps for relative motions.
O Mode	Set **O**perating mode of CY545.
P Pos24	Step to specified absolute **P**osition.
Q	**Q**uit entering commands to external memory.
R Rate	Specify slewing or maximum step **R**ate.
S Slope	Specify acceleration/deceleration **S**lope value.
T Bit,Addr	Loop to address **T**il bit matches value.
U	Reserved command.
V	Reserved command.
W Bit	**W**ait for specified bit to match value.
X	E**X**ecute external memory commands.
Y Addr16	Set external memor**y** address pointer.
Z Cnt16,Addr	**Z**illionLoop to byte address for 16-bit count.
+	Select **CW** direction for relative motions.
-	Select **CCW** direction for relative motions.
/	Negate prefix used with Bit commands.
? Cmd	Query specified command parameter value.
0	Stop execution of commands from memory.
[Addr,Cnt,D1,..,Dn	Special HP-display support command.
"String"	Display all characters between quotes.

Command Formats

The CY545 supports several different command formats, which can be split into two major categories, **ASCII** format and **Binary** format. Most examples in this manual will illustrate ASCII formats, and we believe this will be the most popular mode of operation. The main advantage to Binary format is a savings in the number of characters needed to represent a parameter value, so Binary format commands will usually take fewer characters than ASCII format commands. However, Binary format commands are more difficult for humans to read, and if the CY545 and host computer become unsynchronized in Binary mode, due to a faulty data count, it is extremely difficult to resynchronize the two without a hardware reset.

ASCII Command Format

ASCII format commands all start with the ASCII command letter that selects that command. If the command also has parameters, the command letter must be followed by a space, then the parameter value. Multiple parameters, such as those used by the Loop command, may be separated by a space or comma. Our examples will generally use a comma. All ASCII commands end with the ASCII carriage return character, shown as "<cr>" in the following examples. Commands without parameters, such as the Go command, use only the command letter, followed immediately by the carriage return. For example:

G<cr>	Go command, no parameter
I<cr>	Initialize command, no parameter

Commands with parameters may use two formats for the parameter values in the ASCII mode, a decimal format, consisting of the digits 0 to 9, or a hex format, consisting of the digits 0 to 9, letters A to F, and an "H" suffix. Hex parameters must start with a digit. For example:

R 135<cr>	Rate command, single decimal parameter
R 87H<cr>	Same parameter value, but in hex form
R ABH<cr>	Illegal parameter, no digit to start
R 0AB<cr>	Illegal parameter, no H suffix
R 0ABH<cr>	Parameter value ok
D 12345<cr>	16 bit parameter value in decimal
D 1A3BH<cr>	16 bit parameter value in hex
L 37,27<cr>	Loop command, multiple parameters
L 23H,27<cr>	Parameter formats may be mixed

Parameter values may be entered with any number of digits, but the command will limit the value to a single byte if the range of the parameter is only one byte. For example:

R 0ABCH<cr>	Value is taken as 0BCH = 188
R 350<cr>	Value is taken as 94 = 350 - 256

Sixteen bit parameters are limited to 65535 (0FFFFH), and eight bit parameters are limited to 255 (0FFH), while the 24 bit parameters, used for position and step counts, are limited to 16777215 (0FFFFFFH). All arguments are unsigned integers.

Binary Command Format

The Binary format still starts with the ASCII command letter for each command. However, the format for parameters is quite different in Binary mode. The command letter is always followed immediately by a binary data count, representing the number of data bytes needed by the command. For commands without parameters, such as the Go command, the data count is a binary value zero (not the ASCII character “0"). For commands with parameters, the data count represents the number of bytes needed to specify the parameter(s). This byte count must match the expected size of the parameters. For example, the Number command must always have a data count of three, for its 24-bit parameter, even if the most significant byte values are zero.

The Quit command, which ends the entry of commands into external memory, should not be followed by any data count. This command is immediately followed by the ASCII letter of the next desired command. Also, the Stop command must use the binary value zero, not the ASCII character “0", followed by a data count which is also zero.

All multi-byte parameters must be entered least significant byte first. For example, the Zloop command has a 16-bit count, followed by an 8-bit address, so it has a data count of three, with the data byte sequence being the LSByte of the count, the MSByte of the count, then the branch address. More examples are shown below. The data bytes are shown as hex values, with a space separating each one for clarity, but they are issued to the CY545 as single 8 bit values, without the spaces.

G 00	No argument, byte count zero
R 01 87	Single byte argument
D 02 3B 1A	Two byte argument, value 1A3BH
N 03 11 22 33	Three byte argument, value 332211H
Z 03 21 04 92	Multiple arguments, Z 0421H,92H

The following sections describe each of the CY545 commands. The commands have been organized into groups of related functions.

CHAPTER 5 - MOTOR CONTROL COMMANDS

Motor Control Commands

The motor control commands are used to set the operating conditions for running the stepper motor, and to actually move the motor. Various commands set relevant parameter values, and two commands cause actual motor motion. In addition, the JOG signal may be used to move the motor manually at any time.

Set Stepping Rate

Three commands set up the basic step rate functions for the CY545. They are:

R Rate	Set maximum step rate
F Rate	Set initial step rate
S Slope	Set acceleration slope

The rate parameters for **Rate** and **FirstRate** are single byte values that select the initial and maximum step rates for the motor. These parameters are pointers into the CY545 Rate Table, shown in a later section of this manual. The actual step rate in steps per second comes from a combination of the Rate Table and the operating frequency of the CY545. The step per second values of the Rate Table have been computed for a crystal frequency of 12 MHz. The use of other frequencies causes a linear scaling of the rates, and may be derived by multiplying the given rates by $F_{cy}/12$, where F_{cy} is the actual crystal frequency (in MHz) used for the CY545.

The slope parameter for the **Slope** command determines how quickly the CY545 accelerates and decelerates while stepping. It is also a single byte value, and should range from 1 to 255. Slowest accelerations are selected by small values of slope, while fast accelerations use larger values for slope. In practice, you must experiment with the slope parameter value to determine the optimal acceleration for your specific application. Initial values in the 175 to 225 range might be tried, with adjustments made as required.

Acceleration and deceleration are symmetrical, and are controlled by the single slope parameter. For small values of slope, the CY545 can take a very large number of steps to go from the initial rate to the maximum rate. If the distance to travel is too short to allow the CY545 to reach the maximum rate, it will perform a partial acceleration, reaching the highest possible rate under the current slope parameter, before having to decelerate to stop at the target position.

Set Count and Direction

Three commands set relative mode values for the CY545. They are:

N Num24	Set number of steps
+	Select CW direction
-	Select CCW direction

The **Number** command specifies the number of steps the CY545 is to take in the relative step mode. It uses a 24-bit parameter, so the CY545 can travel + or - 16 million steps from its current position in the relative mode. No stepping occurs when the Number command is issued, the command simply defines the number of steps to take, and this value is stored in an internal N register of the CY545.

The **+** command selects the clockwise direction for stepping. This command causes the CCW signal to go low.

The **-** command selects the counter-clockwise direction for stepping. The CCW signal will go high when this command is used. Also, stepping direction may be controlled externally, by driving the CCW pin with the desired level. If this function is used, the - command should be issued, so the CCW signal is high. It can then be pulled low externally to force clockwise stepping.

If the CCW pin is being driven low by the CY545, it is a stronger sink device, and will require a very strong external signal to force the pin into the high state for CCW stepping. However, if the CCW pin is held high by the CY545, from a - command, it can easily be driven low for CW stepping.

Set Step Mode

Two CY545 commands control step operation modes for the CY545. They are:

A Pos24	Set current step position
C	Select continuous step mode

The **At** command sets the current position kept by the CY545 to the value of its parameter. This number is a 24-bit value, so any position from 0 to 16777215 may be specified. Normally, this command is used after stepping to a known mechanical position, such as a “home” position sensor, and can calibrate the CY545 to make this position always be the same number.

While the CY545 is stepping, the internal current position is always updated. The position is incremented during CW stepping, and decremented during CCW stepping. You may query the CY545 for its current position while it is not stepping.

The position value is not too important for relative mode stepping, in which the direction and number of steps are specified. However, it is very important for absolute mode stepping, in which a target position is specified, and the CY545 computes the direction and number of steps required to reach the target.

The **Continuous** command sets a special step mode in the CY545. In this mode, the CY545 will accelerate from the initial rate to the final rate, and then run continuously. It will not ramp down or stop at any particular position or number of steps.

In order to stop a continuous motion, the INHIBIT-ABORT signal must be pulled low, forcing the CY545 into a deceleration mode, and then be kept low to stop the CY545. The CY545 will stop, after it has decelerated to the starting rate, if the INHIBIT _ABORT signal is still low.

The continuous mode is set by the C command, but stepping does not begin with this command. The continuous mode may also be selected by external signal. This is done by forcing the SLEW signal low as the CY545 starts stepping.

Moving the Motor

Finally, two commands are used to actually make the motor move, based on the currently defined parameters. They are:

G	Step in a relative mode
P Pos24	Step to the specified target position

The **Go** command selects relative mode stepping. The CY545 will step based on the last Number of steps specified and the selected step direction. A continuous mode step motion may also be started by this command.

Absolute mode stepping is selected by the **Position** command. This command both specifies the desired target position and starts the motion. When the target is specified, the CY545 will compute the number of steps to take and the direction of motion, based on the current position register.

If the target position is greater than the current position, the CY545 steps in the CW direction. If the target position is less than the current position, the CY545 steps in the CCW direction.

Recall that the current position is defined by the At command, and is updated with every motion. It is a 24-bit value, ranging from 0 to 16777215. The Position command limits stepping to a number of steps within this range.

Position Counter "Wrap-around"

If a relative motion causes overflow or underflow of the current position register, an absolute move would now work within the new block of 16 million steps. For example, if the current position is 00000020, and the following commands are executed:

N 30<cr>

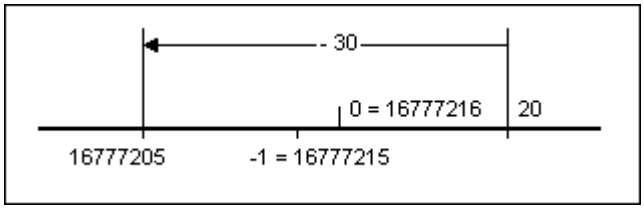
-<cr>

G<cr>

The motor will step 30 steps CCW, and the current position register will underflow to a value of -10, which is 16777205. If you now command:

P 0<cr>

The motor will step 16 million steps in the CCW direction, since the CY545 believes the motor is at position 16777205. It does not remember that an underflow occurred, and will not take 10 steps CW to get back to the original position 0.



This behavior also allows stepping to regions beyond the 16 million step block supported by the CY545. As long as you can work within a block of 16 million steps, relative mode stepping can be used to move from one block to the next, with the host computer keeping track of which block the CY545 is currently located in.

CHAPTER 6 - BIT FUNCTION COMMANDS

Bit Function Commands

Several commands allow you to specify which User Selectable Bits are used by the command. The CY545 has at least six general purpose signals available, even if all hardware support options are used. More bits are available when some hardware support, such as the Hewlett Packard LED display, the CY233, or the parallel interface are not used. Each of these commands includes a parameter which specifies which bit or combination of bits to use with the command. The specific values are discussed with the relevant commands.

User Bits

One command allows you to set or clear specified bits. It is:

B Bit Set or clear specified bits

When in the ASCII mode, this command also allows a special format, using a complement prefix to mean the inverted value of the specified bit parameter. The prefix is the "/" character, and is used as follows:

B 2<cr> Set User Selectable Bit 2 high

/B 2<cr> Clear User Selectable Bit 2 low

The same function may be performed by the bit parameter value. When a "/" character is detected, it simply complements the fourth bit of the bit value parameter, so the same functions as above can be specified by:

B 02H<cr> Set Bit 2

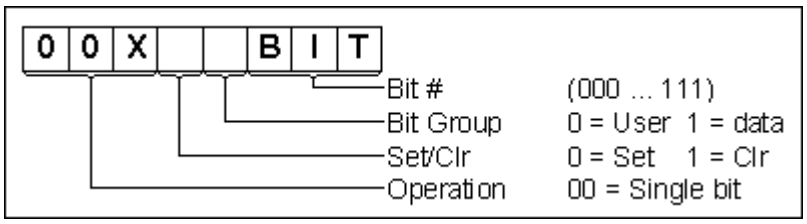
B 12H<cr> Clear Bit 2

The "/" character is equivalent to performing an exclusive OR (XOR) of the bit number parameter and the value 10H (16 decimal).

The value of the Bit parameter will determine what action to take, as follows:

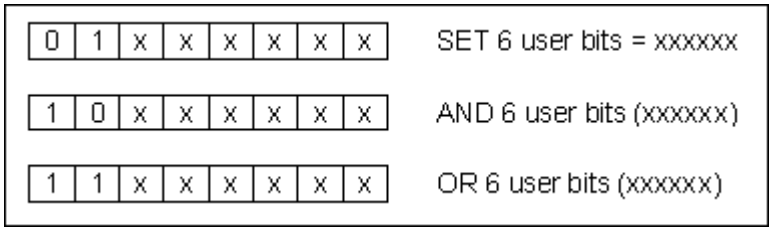
00H to 07H	(0-7)	Set one of user bits 0 to 7
08H to 0FH	(8-15)	Set one of data bus bits 0 to 7
10H to 17H	(16-23)	Clear one of user bits 0 to 7
18H to 1FH	(24-31)	Clear one of data bus bits 0 to 7
20H to 3FH	(32-63)	(Same as 00H to 1F)
40H to 7FH	(64-127)	Set user bits 0 to 5 at one time
80H to 0BFH	(128-191)	AND user bits 0 to 5
0C0H to 0FFH	(192-255)	OR user bits 0 to 5

As seen from the above values, the bit argument is composed of an "opcode" part and an "operand" part. The general coding of bit arguments is shown below:



The 00 opcode specifies that one bit is to be operated on. The three least significant bits identify the bit, while two other bits specify the bit group (user or data bus) and the operation (set or clear). One argument bit is not used in single bit operations.

If the opcode is non-00, then all six bits are operated on as a group, with the relevant opcodes shown below.



The actual application will constrain the number of User Selectable bits that are available. When all features of the CY545 are used, only User Selectable bits 0 to 5 are available for general purpose functions. If the CY233 is not used, or the parallel Hewlett Packard LED display is not used, bits 6 and/or 7 will also be available. Finally, if the parallel command interface is not used, and there is no local memory or HP LED display, the CY545 data bus lines will also be free, and can be addressed as bits 8 to F of the Bit command. These special function lines can only be tested or changed on an individual basis.

> ***NOTE:*** *Setting or clearing these Special Function bits when associated hardware is present could cause unwanted interactions between the CY545 and the hardware. Also, the CY545 will change the values on these signals as required for normal support of the hardware. Be sure to manipulate only those signals that are available to the application.*

Since **User Selectable bits 0 to 5** will always be available to the application, they can be treated as a group, rather than as individual signals. The most significant two bits of the Bit command parameter value indicate this selection. If any of these two bits is a 1, the lower 6 bits of the value are treated as a group to modify USRB0 to USRB5 all at once. Options are to set all six bits at once, as specified by the parameter value, to logical AND the parameter value with the six bits, and to logical OR the parameter value with the six bits.

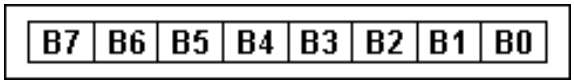
Bit Dependent Jump and Wait

Two CY545 commands use the bits for special testing functions. They are:

T Bit,Addr	Jump to address until bit matches value
W Bit	Wait for bit to match value

For these commands, the bit value parameter is treated as follows:

00H to 07H	Test one of user bits 0 to 7 for 1
08H to 0FH	Test one of data bus bits 0 to 7 for 1
10H to 17H	Test one of user bits 0 to 7 for 0
18H to 1FH	Test one of data bus bits 0 to 7 for 0
80H to 0BFH	Test user bits 0 to 5 for a group match



Both commands compare the value of the I/O signal to the value specified by the parameter. Note that **USRB0 to USRB5** can be treated as a single group of six bits, with the value test comparing all six bits to the lower six bits of the parameter. Any other bits can only be tested on an individual basis.

The **Til** command will jump to the specified byte address until the signal(s) match the parameter value, either a low or high level. When there is a match in value, the Til command does not take the jump, and execution falls through to the command after the Til command.

The **Til** command is only used in an external memory program, since it will jump or fall through, depending on the bit value. The address of the jump is a single byte, limiting the jump to an address on the same 256 byte page as the command after the Til command. The TIL command may jump to itself.

The **Wait** command simply waits for a match of the bit(s) against the parameter value. When there is no match, the CY545 stays in the Wait command, and continually tests the signals until a match occurs. Once a match is detected, the CY545 will continue with the next command.

If you issue a new command either in parallel or serial format while the CY545 is stuck in a Wait command, the CY545 will abort the **Wait** function and act on the new command. This is a mechanism for breaking out of Wait commands that are stuck because the proper bit pattern does not occur.

The **Wait** command is normally used in an external memory program, to synchronize the CY545 with some external control or status signal. For example, move a part into place, activate some operation by changing a user bit, wait for a signal back from the operation indicating operation complete, then remove the part.

The example illustrates the use of **Wait** and **Til** commands for the operation described above:

```
Y 100<cr>          Program start address.
E<cr>              Next instructions go into memory.
W 1<cr>            Wait til USRB1 = 1.
+ <cr>
G<cr>              Go (200 steps by default).
/B 2<cr>           Drive USRB2 = 0 as signal.
W 11h<cr>          Wait til USRB1 = 0.
– <cr>
G<cr>              Return same number of steps.
B 2<cr>            Drive USRB2 = 1 again.
T 10h,100<cr>      Test USRB0 & jump unless it is 0
0<cr>              then fall thru and stop the program
                   by returning to command mode.
Q<cr>              End of program definition.
Y 100<cr>          Reset the address.
X<cr>              eXecute the above program.
```

Step/Test/Seek Home Command

Finally, one command uses the bits to control special motions, and cause stepping during the bit testing. It is:

H Bit Step to Home, using specified signal

The values of the parameters for this command are treated the same as the **Til** and **Wait** commands. That is, for the **Home** command, the bits are tested for a match to the parameter value.

As the CY545 steps the motor for this command, it will run at a constant step rate, with no acceleration or deceleration. The desired step rate is chosen by the First Rate parameter, set by the F command. The actual step rate is a derivative of the rates in the **CY545 Step Rate Table**. For the **Home** command (and the Jog pin function), the CY545 will divide the selected rate by 20. When operating at 12 MHz, this gives a range of about 1 step per second to about 1000 steps per second for this command.

The **Home** command starts by testing the specified bit pattern for a match with the external signals. If there is a match, the CY545 will begin by stepping in the CCW direction, and will continue in this direction until there is no longer a match.
When there is no match between the bit parameter and the external signals, the CY545 steps in the CW direction. It will continue to step in this direction until the signals match the parameter value. At this point the CY545 stops, and sets the current position to zero.

Thus, the **Home** command steps the motor to seek the edge between a match and no match condition on the home signal. The CY545 then stops at the first step where a match occurs, and is always stepping in the CW direction looking for this match. This mechanism should always seek the same mechanical position, since any directional backlash is compensated by using the same CW step direction in seeking the final home position.

If the CY545 runs into one of the limits while looking for the home signal, it will abort the home command and stop stepping. The current position is not changed to zero in this case.

Special Note Regarding Older Versions

This special note applies only to the chip version prior to the CY545A. The older CY545s drive the BUSY line low at the beginning of the Home command, then leave it low after the home position has been found. This means that the CY545 does not indicate ready for the next command on the parallel command interface.

There are two approaches to solving this problem. First, if the CY545 is displaying the position for every step that it takes, and the position display is going to the parallel interface, the BUSY signal will be driven to the ready state when the CY545 completes the home function. See the Mode command for details on position display during home.

The other approach is to issue the first character of the next command with a blind handshake, and wait for the CY545 to accept the character. With this approach, you should place the command character on the data bus, then drive IO_REQUEST low for 100 usec. Drive it high again after 100 usec, then wait for 500 usec. If the CY545 drives the BUSY signal high, it has accepted this first character, and is ready for the next character of th70e command. Otherwise, repeat the blind handshake until the CY545 drives the BUSY signal high.

While the CY545 is performing the home function, it will ignore the IO_REQUEST signal. However, when it finishes, and sees the blind handshake described above, it will accept the next command character, then drive the BUSY signal high again. At this point it will be ready for a normal handshake for the rest of the command sequence.

CHAPTER 7 - EXTERNAL MEMORY COMMANDS

External Memory Support Commands

Several CY545 commands provide support for the optional external memory. They control the addressing, data entry, and execution of commands from the memory.

External Memory Address Pointer

The first command is used to set the address pointer. It is:

Y Addr16 Set external memory address pointer

The 16 bit argument to the Y command is placed into the memory address register of the CY545. Other commands may use this value for their functions.

The Y command does not change modes or other features of the CY545, so when issued as a direct command, it simply sets up the address pointer. The CY545 will then wait for the next command. However, **when the Y command is executed from external memory**, redefining the address pointer causes the CY545 to execute commands from the new address, so **the Y command becomes a 16-bit jump command**. This dual function allows you to specify the working address when defining the memory contents, and jump to any location of the memory when running from the memory contents.

Writing Commands into Memory

The two commands that control the entry of data into memory are:

E Enter following commands to memory

Q Quit entering commands to memory

The **Enter** command is issued to the CY545 in the direct command mode. Once this command is executed, the CY545 changes to the program entry mode. All following commands will be written to the external memory. The CY545 will not execute any of the commands, it simply writes them into the memory.

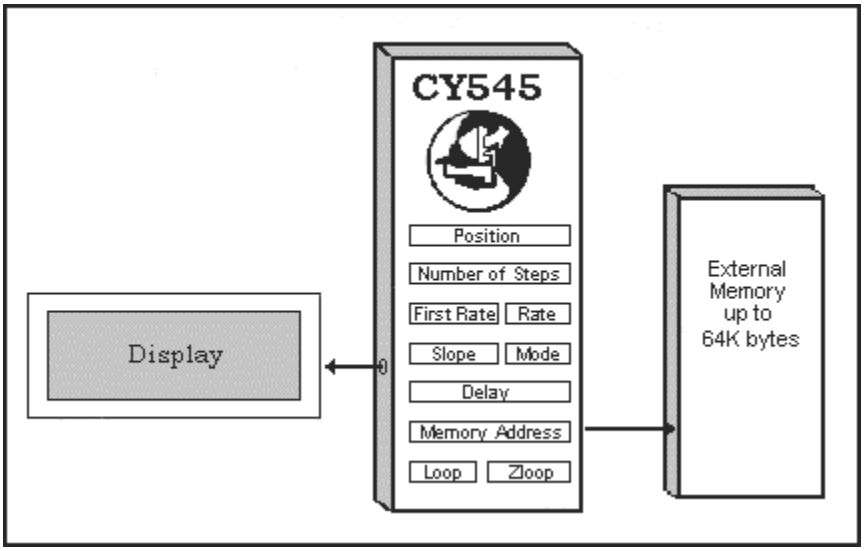
> **Commands will be written to memory starting at the current address set by the Y command, and the address pointer will increment once after writing each command character.**

The memory contents consist of a string of characters, and are identical to the characters issued to the CY545 in direct command mode. Every character sent to the CY545 will be recorded in memory exactly as it is sent to the part.

The memory writing function continues until the CY545 receives a Quit command. It then returns to the direct command mode, in which every command is executed immediately as it is received. The address pointer value now contains the address where the next character would have been written, and the difference between the starting value and ending value will indicate

the amount of memory required for your program. Before running the program, the Y command should be used to redefine the pointer value for the start of the program.

Note that the carriage return character is optional after the Quit command in the ASCII command mode, and a data count is not used after the Quit command in the Binary command mode.



Running From External Memory

Two commands are involved in running programs from the external memory. They are:

X	Execute external memory commands
0	Stop external memory command execution

The **Execute** command changes the CY545 from the direct command mode to the program execution mode. Your program will begin running from external memory, at the current address in the address pointer. The **X** command would normally be preceded by a **Y** command to set the program starting address.

In the program execution mode, the CY545 will begin reading command characters from the memory, incrementing the address pointer after every read operation. When an entire command has been read into the part, that command is executed similarly to directly issued commands.

When the current command is finished, the CY545 will read the next command from memory and execute it. This process continues until the CY545 finds a stop command.

The **Zero** command (ASCII character 0, not binary value zero) is the **stop execution** command. When this command is executed, the CY545 changes from program execution mode back to direct command mode. It then waits for the next command to be issued by the host system. Note that in the Binary command mode, the binary value zero should be used for the stop execution command, followed by a zero value data count.

You may also issue commands to the CY545 while it is running a program from external memory. The CY545 will process these commands between the commands read from the memory. The ability to issue direct commands while the CY545 is running an external memory program provides a degree of "live" interaction, allowing you to change parameters or modify the execution flow by the direct commands. It is even possible to halt a program, by issuing the **Zero** command!

> **Note:** A special feature of the CY545 is the ability to automatically run an external program at power-up, or whenever the CY545 is reset. See the "Auto-Start" feature in chapter 12 on External Memory Support.

Example

From this discussion, you can see that one command controls the basic memory addressing function, while two pairs of commands control the definition of memory contents and execution of memory contents. A simple example is shown below:

F 15<cr> **S 200<cr>**	Define some parameters in direct mode
...	Additional commands are possible here
Y 50<cr>	Set memory pointer to address 50
E<cr>	Start program entry at location 50

The following highlighted code is written to memory, but not executed:

N 400<cr>	First command in external memory
+<cr>	
G<cr?	Take 400 CW steps
D 1000<cr>	Delay for 1 second (1000 msec**)**
N 350<cr>|	Reset number of steps
-<cr>	
G<cr>	Take 350 CCW steps
D 750<cr>	Delay for 0.75 seconds
0<cr>	Stop, last command in external memory
Q<cr>	End of program definition
...	Other direct mode commands may be here
Y 50<cr>	Reset address pointer to prog start
X<cr>	Execute. Begin running the program

The program in the external memory will run after the X command, and will continue until the stop command is found. The program takes two relative motions, with different numbers of steps, and in different directions. When the Stop command is seen, the CY545 goes back to direct command mode and waits for the next command. The program could be repeated by sending the Y and X commands again, since the program continues to reside in the external memory.

If non-volatile external memory is used, such as an EEPROM, the program will be stored in the memory even when power is removed from the CY545 system. So, once a program is defined, it may be used over and over, by simply pointing to the program address and sending the Execute command.

More hardware details about external memory support are provided in a later section.

CHAPTER 8 - PROGRAM BRANCH COMMANDS

Jump and Loop

Several CY545 commands control program branching and flow while the CY545 is executing a program from external memory. These commands normally are used only within a program, not as direct CY545 commands. The CY545 will however execute the commands in the direct mode, but the result may be different than if the commands were run from a program. The commands are:

J Addr	Jump to byte location of current page
L Cnt,Addr	Loop to byte address for count times
Z Cnt16,Addr	Zloop to byte address for count times

In addition to the above commands, the Y command, when executed from a program, provides a 16 bit jump that can reach any address of the external memory.

The above listed commands all use an 8 bit address value. This restricts the range of addresses to the 256 bytes of the current memory page. When a jump is taken, the lower byte of the current address pointer is replaced by the 8-bit address parameter value, and program execution continues at this new 16-bit address. If one of the above commands lies on a page boundary, the address pointer is incremented before the command is executed, so the branch would be taken into the new page of the memory, not the previous page that started the command.

The **Jump** command simply provides a short version of the branch function. It can reach any address on the current memory page.

The **Loop** command provides a repeated branch, with a specified number of iterations. Both parameters are byte values, so a loop may be repeated 255 times, and the branch must be to an address on the current page.

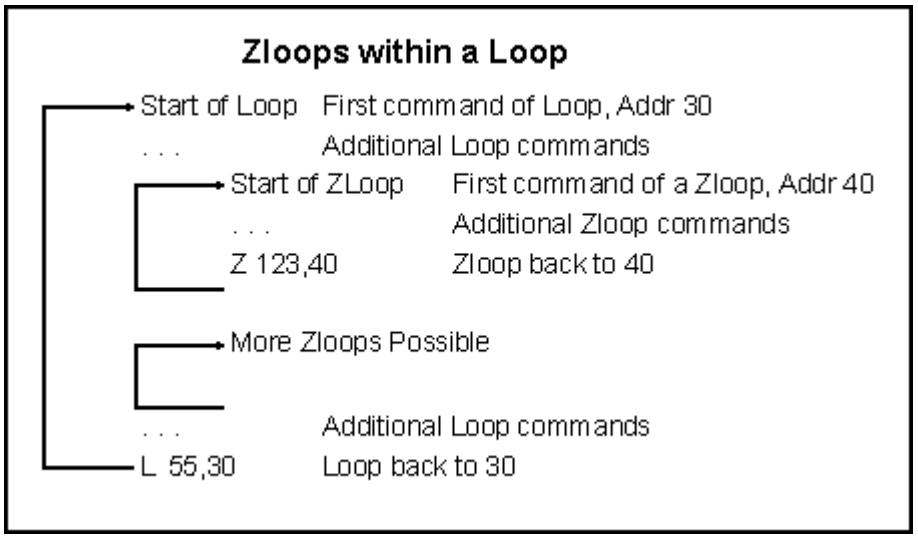
The **Zloop** command also provides a repeated branch. However, the repeat count is a 16-bit value, so the Zloop may be repeated up to 65535 times. The branch is still restricted to an address on the current page.

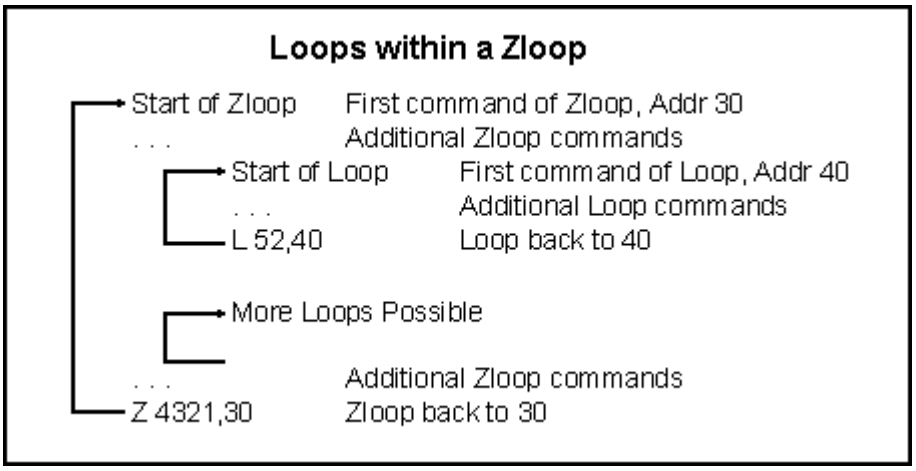
For both loop commands, the instruction following the loop is executed when the loop count is exhausted.

Nested Loops

The combination of the Loop and Zloop commands provides two levels of nesting to loops. Only one of each type of loop may be active at a time. That is, a number of loops may reside within one zloop, or a number of zloops may reside within one loop.

However, one loop may not reside within another loop, and one zloop may not reside within another zloop. This is shown below:





Recall that the CY545 also supports a branch command using the bit instructions, the Til command. It tests the I/O signals against the specified bit parameter value, and jumps if the values do not match. Otherwise execution continues with the instruction after the Til command.

CHAPTER 9 - OPERATING MODE COMMAND

Operating Mode Command

Another command is used to define the Operating mode of the CY545. It is:

O Mode Set CY545 operating mode

The parameter of the Mode command is stored in an 8-bit internal CY545 register. These bits are only accessible by the Mode command, not through any external I/O lines, such as the User Bits. Each bit of the Mode command register controls a CY545 feature, as shown below:

MB7 Select ASCII command mode if set, else Binary
MB6 Select serial baud rate if set
MB5 Enable CTS/ signal on USRB6 if set (CY545A and later)
MB4 Display position during Jog & Home cmd if set
MB3 reserved
MB2 Select HP Display if set & MB0 = 1
MB1 Select CY325 based display if set & MB0 = 0
MB0 Select parallel display if set, else serial

The **Mode** command will define the values for all 8 bits of the register at once, so you must be careful not to affect the value of one feature while changing the value of another. The default value for the mode register is 80H at power up. This selects the ASCII command mode with

MB7	MB6	MB5	MB4	MB3	MB2	MB1	MB0
1	0	0	0	0	0	0	0

= 80h, default mode

The most significant bit selects between ASCII command mode and Binary command mode. When the bit is set, ASCII mode is selected, and when the bit is clear, Binary mode is selected. When you use Binary command mode, you must first issue an O command in ASCII mode that turns off the upper bit, to put the CY545 into Binary mode. All commands following the O command are then in the special Binary format:

O 0<cr> ASCII command to select Binary mode
... All following commands in Binary format

When MB4 of the mode register is set, the CY545 will display the current position while stepping in the Home command or while stepping from a JOG signal input. This position display may have some effect on the step rate of the command, since additional time is required during each step to display the position.

The lower three bits, MB0, MB1, and MB2, are used to select what device will be used to display query outputs and string value outputs from the CY545. Two basic display choices are a serial display or a parallel display. The least significant bit, MB0, selects between these two options.

When a serial display is selected, the next bit, MB1, selects between a standard serial output, used for connections to a host computer serial port or general terminal serial port, or a specialized LCD display device, controlled by the CY325 LCD Display Controller.

When bit MB1 is zero, a standard serial display is used, and the CY545 does not perform any special formatting of the output. This is the power-on default option.

When MB1 is one, the CY325 based display is used, and the CY545 adds special formatting strings to the data, putting the query results in fixed places on the display screen. This mode should not be used without a CY325 controller. For more information about this device, contact Cybernetic Micro Systems or visit our web site at www.controlchips.com.

When a parallel display is selected by MB0, MB2 selects between a standard parallel output, and a special Hewlett Packard LED display.

MB7	MB6	MB5	MB4	MB3	MB2	MB1	MB0	
				x	x	0	0	Standard Serial
				x	x	1	0	CY325 Serial
				x	0	x	1	Standard Parallel
				x	1	x	1	HP LED Parallel

When MB2 is zero, the standard parallel output is selected. This uses the IO_REQUEST and BUSY signals to control the reading of information from the CY545. Also, the CY545 will drive the FPL signal low (shared function with USRB6) when it has something to output. This protocol will work with a standard two-line handshake or with the parallel functions of a CY233 Network controller.

When MB2 is one, the special HP LED display is selected. In this case, all output from the CY545 goes to the local HP LED display, and a special signal, HP_SEL (shared function with USRB7), is generated while the CY545 is writing to the display. More information about support for this display is provided in a later section.

When a Mode command is issued with MB5 set to one, the CY545A enables the CTS/ function on the USRB6 (the CTS function is not available on older parts prior to the CY545A). After any reset, this function is disabled, and USRB6 is high, so the CTS/ signal is off. A Mode command must be issued to the CY545 before the CTS/ function is enabled.

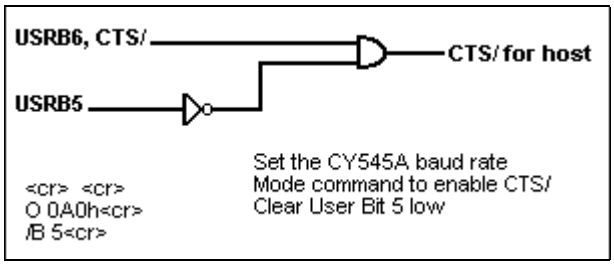
When the signal is low, the CY545 is ready for more serial characters, and when the signal is high, the CY545 is busy processing a command. The CTS/ signal can be used to automatically control the rate of serial communications from the host system, stopping transmissions while the signal is at TTL one. This allows the host to send a sequence of commands, including motion commands that may take some time to execute, without worrying about the CY545 being able to accept the commands. The hardware control insures that the CY545 will always indicate when it is ready for more serial data. The exception to this is during use of the external jog line, in which CTS is not set busy during Jog stepping.

Since the CTS/ signal is initially disabled, the CY545 must receive a Mode command that enables the signal before it can be used by the host. There are three alternatives for turning on the CTS/ function.

First, the Mode command could be read from the local memory, if one is present, as part of the "Auto-Start" routine. This is probably the best approach, since the CY545 will always reconfigure itself after a reset, without any special action by the host. With this approach, CTS/ can automatically be enabled by the CY545 itself. See the section on External Memory Support in chapter 12 for more details on the "Auto-Start" function.

If your system does not have the external memory, an alternative approach is to send the Mode command from your host computer. However, this requires some special software, since the CTS/ signal is off when the CY545 is reset. The software must send the Mode command with the CTS monitor function disabled, so that your computer will transmit the command even though CTS/ is off. It can then reconfigure itself to enable CTS monitoring, after sending the Mode command. An example BASIC language CRT emulator program which performs this function is shown in chapter 20 (Getting Up and Running).

The third alternative for starting the CY545, with CTS/ enabled requires some external hardware plus another User Bit. With this design, the CTS/ signal is always enabled, so the host computer is not required to reconfigure itself, first sending commands without CTS monitoring, then switching to CTS monitoring. When the CY545 is reset, both the CTS/ signal (USRB6) and the second User Bit signal, say USRB5, are high. This forces the CTS/ output from the AND gate to be low, which is the active state. The host computer can now send the Mode command to enable the CY545 CTS/ function, then it must send a Bit command to drive the second User Bit to zero. The logic will now follow the value of the actual CY545 CTS/ signal, until the CY545 is reset again. For host systems that always require a CTS/ signal, this alternative will work when there is no local external memory from which to automatically execute the Mode command, The circuit and command sequence are shown below:



Finally, if MB6 is one, the CY545 uses the lower four bits of the mode byte value to select the serial baud rate. This allows a parallel command to set the serial rate to a desired value, without tying the I/O Request and Busy lines to a particular state, and allows a parallel command source to set up the CY545 for serial communications, such as showing messages on a serial display.

For this function, the CY545 does not store the mode byte value in its internal Mode register, so if MB6 is set, no other operating modes of the CY545 are affected, only the baud rate selection. Bits MB0 to MB3 are treated as a binary value, selecting the baud rate as shown in the table below:

MB 3	MB 2	MB 1	MB 0	MB value	Selected Rate @ 11 MHz
0	0	0	0	0	300
0	0	0	1	1	600
0	0	1	0	2	1200
0	0	1	1	3	2400
0	1	0	0	4	4800
0	1	0	1	5	9600
0	1	1	0	6	19200
0	1	1	1	7	57600
1	0	0	0	8	Self Adaptive (up to 19200 baud)
1	x	x	x	9-F	Undefined

When crystal frequencies other than 11 MHz are used, the fixed rates will be scaled linearly by the crystal frequency, while the self-adaptive rate will attempt to compensate for the new crystal frequency. For most crystal frequencies above 11 MHz, the CY545 can adapt successfully to baud rates of 4800 baud or less, with a small enough error to insure reliable operation of the serial interface. However, at 14.7456 MHz, the CY545 can adapt to all baud rates up to 19,200, plus two non-standard rates of 38,400 and 76,800 baud.

The CY545 will set the baud rate during the execution of the Mode command, and the new rate will be in effect until a hardware reset occurs or another Mode command is issued, with B6 set. When the adaptive rate is selected, the CY545 must receive two carriage returns before the operating baud rate will be defined again, so this selection may only be used if serial data is coming into the CY545. No transmissions will occur until the baud rate is again defined.

CHAPTER 10 - MISCELLANEOUS COMMANDS

Initialize Command

Finally, the CY545 has several miscellaneous commands for general control purposes. One command provides a software reset for the CY545. It is:

I Initialize CY545

When the **Initialize** command is processed, it acts like a power-on reset. The CY545 command mode is changed to direct command mode, all I/O signals are brought high, and the serial baud rate is reset. Any program that was executing from external memory is stopped if an **I**-command is read.

The **Initialize** command and the power-on reset will redefine the CY545 stepping parameters to the following default values:

R 100
F 14
S 220
N 200

and the current position register is cleared to zero.

The current values of these registers will always be changed back to the defaults when a reset is performed or an I-command is executed.

Time Delay Command

The CY545 also provides a time delay function, with:

D Del16 Delay for specified milliseconds

The parameter for the **Delay** command is a 16-bit value, calibrated in milliseconds when running at 12 MHz. This command simply causes the CY545 to wait for the specified time before going to the next command.

When used in an external memory program, the command allows you to pause for a known time between program functions. For example, take a specified motion to move a part into place, then delay for 3 seconds while some operation is performed, then remove the part.

The value of the Delay parameter allows a time delay between 1 millisecond and 65 seconds when the CY545 is running at 12 MHz. Longer delays can be created by repeated delay commands, or delay commands in a loop.

Also note that very slow stepping is possible by a loop that consists of a single-step relative motion followed by a time delay. For example:

Y 10<cr>	Select address 10 to start code
E<cr>	Enter to memory

N 1<cr>	Set to 1 step relative
+<cr>	Select CW direction
G<cr>	Take a step
D 2000<cr>	Delay 2 seconds
L 50,16<cr>	Loop back to G command, 50 times
0<cr>	Stop program

Q<cr>	Quit entering program to memory
Y 10<cr>	Reset address
X<cr>	Run the program

The above example will take 50 steps, with a delay of two seconds between steps. Other delay values will provide different step rates.

Query Command

Display CY545 Status

Along with display strings, the CY545 allows you to query the values of various working parameters. The command is:

? Cmd Query specified parameter value

The “Cmd” portion of the **Query** command is a single letter, selecting the parameter for display. Possible choices are:

N	Number of Steps value
P	Current position value
R	Current final rate value
F	Current first rate value
S	Current slope value
Y	Current memory address pointer value
B	Current user bits input value

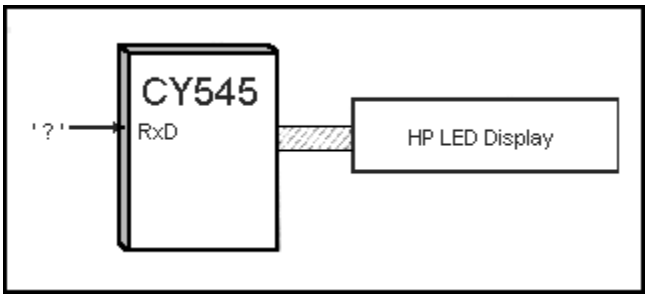
When the Query command is received or read from memory, the CY545 displays the parameter value as follows:

X=vvvvv<cr> ie: **S=00220<cr>**

Where “X” is the parameter letter, and “vvvvv” is the 5-digit ASCII decimal value for the parameter. Parameters that only have 8-bit values will have leading zeros for the unused digits. This is a fixed format output while the CY545 is in the ASCII command mode. Since Position and Number of steps are 24-bit values, these two queries will output an 8-digit number, not the standard 5-digit number.

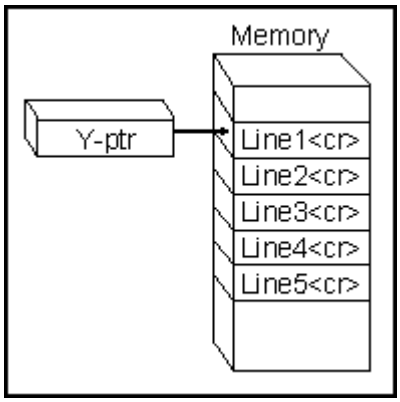
In the Binary command mode, the “vvvvv” portion is replaced by two bytes that represent the binary value of the parameter, sent most significant byte first. The carriage return is not sent in the Binary command mode. Position and Number are sent as three byte values.

The parameter value message is sent to the currently selected display device, so it is possible to send the result to something other than the command source. For example, a host computer could issue the Query command over the serial interface, with the CY545 displaying the result on a local HP display.



Each query command will output the current value of the requested parameter. For the user-bits query, a 16-bit value is returned, with the lower 8 bits corresponding to the CY545 User Bits, and the upper 8 bits corresponding the the CY545 Data bus.

Display Memory Contents



One more parameter may be issued for the Query command, the “M” parameter. The format for this form is special, because it also includes a second parameter, which is a count value. For example:

? M,5<cr>

For this Query command, the “M” is selecting a query of the external memory, and the count indicates how much to display.

The memory display starts by showing the value of the memory address pointer, since the query will start at this address. This is followed by the number of lines specified by the count value. Each carriage return read from the memory counts as one line. This format operates in the ASCII command mode.

In the Binary command mode, you may still perform a memory query, and the query command takes two parameters, the “M”, and the count. However, in this case, the count is the number of bytes to read from the memory, not the number of lines.

Message Display Command

The CY545 also provides a mechanism for displaying messages that you define. This is done by:

"String" Display all characters between quotes

Normally, these messages would be part of a program in external memory, and can announce specific items on the display. The messages can also be prompts to an operator to turn something on or off at particular points in a program, with the CY545 then delaying, or waiting for a signal before continuing with the program.

The string format is very general. Any character is allowed between the quotes (except another quote), including ASCII control codes and carriage returns. The only character that terminates a display string is the double quote character (ASCII 22H), also used to start the string. This is the only character that may not be part of the string.

Strings are stored in external memory if they are received following an Enter command, just as regular CY545 commands. Every part of the string, including the starting and ending quotes is saved. The next CY545 command immediately follows the ending quote of the string. In Binary mode, strings do not include data counts, they are treated like ASCII mode for the duration of the string.

Embedded Control Characteristics and Their Use:

The string command allows any 8-bit characters (except ") to be transmitted as part of the string. This will allow embedded control codes and special "setup" commands to be issued to peripheral devices, such as the CY325 LCD windows controller. For example, the string:

"<Ctrl-C>V 0D6h<cr>
<Ctrl-D>Hello<cr>"

puts the CY325 into the command mode (Ctrl-C character), then issues a Viewport command, which defines a currently active window for display. The window selected uses one quarter of the 711 display, from the middle. The Ctrl-D character then places the CY325 back into the display mode, and the message “Hello” is written to the selected window. The carriage return advances the cursor to the line directly below the “Hello” message.

The above sequence is sent as one string message by the CY545, since all the commands and control codes are embedded within one character sequence, between the double quote characters.

Hewlett Packard (HP) LED 8 Digit Display

Finally, the CY545 has a special command for controlling the Hewlett Packard display. The command is:

[Addr,Cnt,D1,...,Dn Special Hewlett Packard display support

This HP Display command allows you to manipulate some special features of the HP display, such as brightness control or custom character fonts. Use of this command is optional, since it changes the defaults as set by the CY545. It is not required for normal operation of the HP display.

LED Display Details

The address parameter is the display address used to write the data and control information. For example, 30H is the address for the HP control register. The count parameter is the number of data bytes to write, starting at the specified address. If the count is zero, nothing is written, but the internal display address is changed to the specified first parameter value. When the count is non-zero, the following bytes are written to the display, with the count determining how many bytes are written.

One address is treated specially by this command, address 30H, for the control register. If address 30H is followed by a zero data count, the CY545 default control value will be selected. Otherwise, the value of D1 will become the new control register value, and can be used to change the display intensity or turn on the blinking feature.

For more information about the HP display and CY545 support, see a later section on Display Support and the HDSP-211x data sheets (available from Hewlett Packard).

Also note that the HP display is limited to 8 characters, so the Query output format automatically changes when the HP display is selected, and the Position or Number parameter values are queried. In these cases, only the 8 digit value is shown, not the parameter letter and equal sign. For example, instead of showing:

N=12345678<cr>

as it would for a normal serial or parallel display, the CY545 simply shows:

12345678<cr>

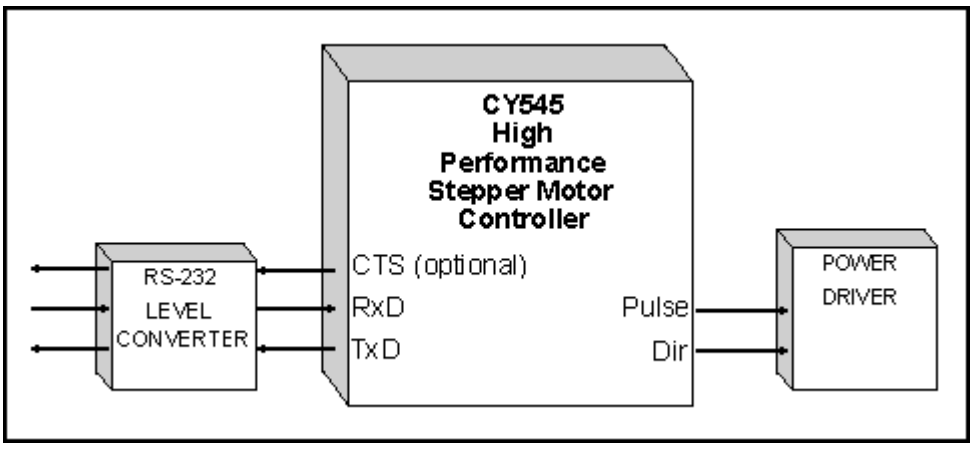
when the ? N or ? P commands are issued for the HP display.

CHAPTER 11 - CY545 EXAMPLE CIRCUITS

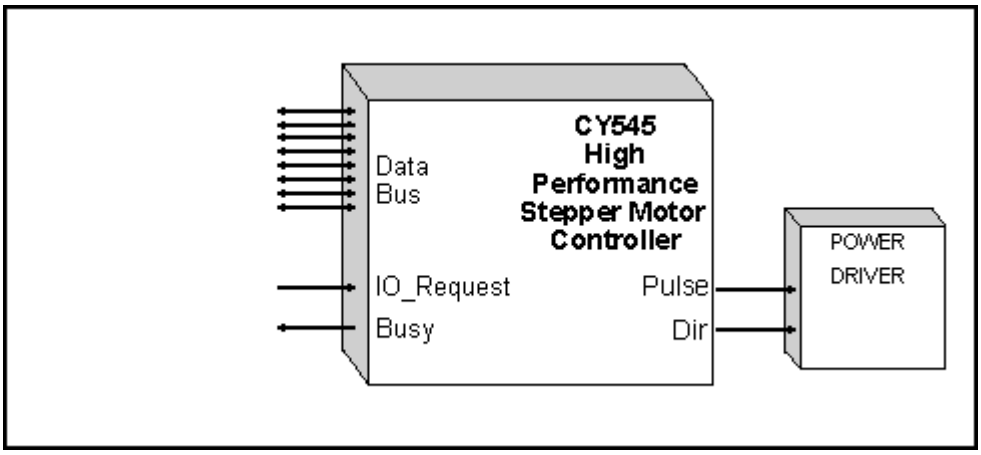
Three Example Cases

The three following circuit diagrams illustrate the wide choice of possibilities for CY545 applications circuits. In the first, only the serial command interface and the pulse and direction lines are used. This represents a minimal CY545 design. The second example is similar, but uses the parallel input bus to send commands to the CY545, with the pulse and direction lines still sent to a power driver. The third diagram illustrates a full blown system with RS-232 serial input, external memory, display circuits, thumbwheel interface, and I/O lines, and is implemented on the CYB-545 prototyping board.

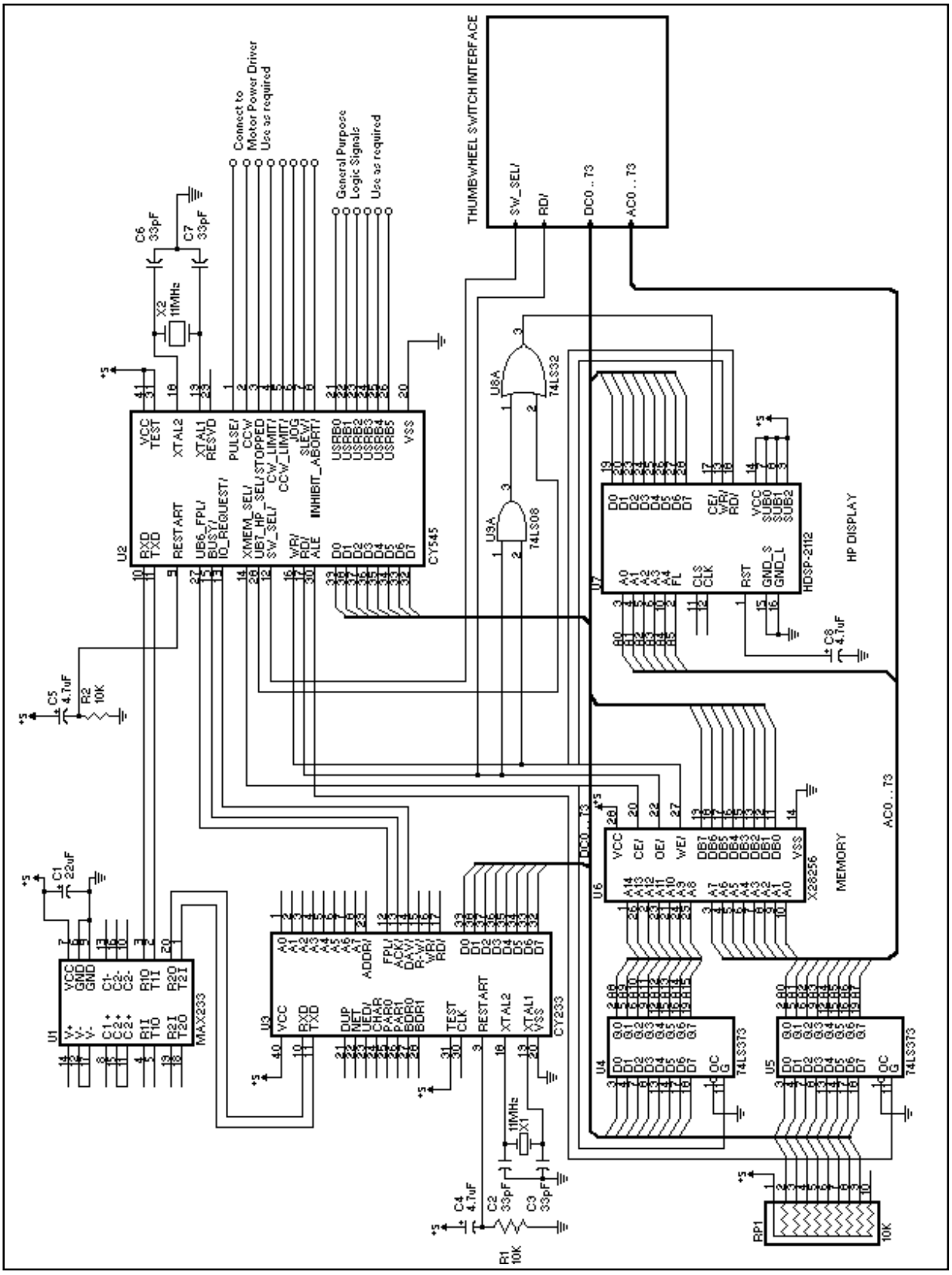
CY545 Example Circuit – Minimum Serial Configuration



CY545 Example Circuit – Minimum Parallel Configuration



CY545 Example Circuit – Maximum Configuration



CHAPTER 12 - EXTERNAL MEMORY SUPPORT

Memory Read/Write

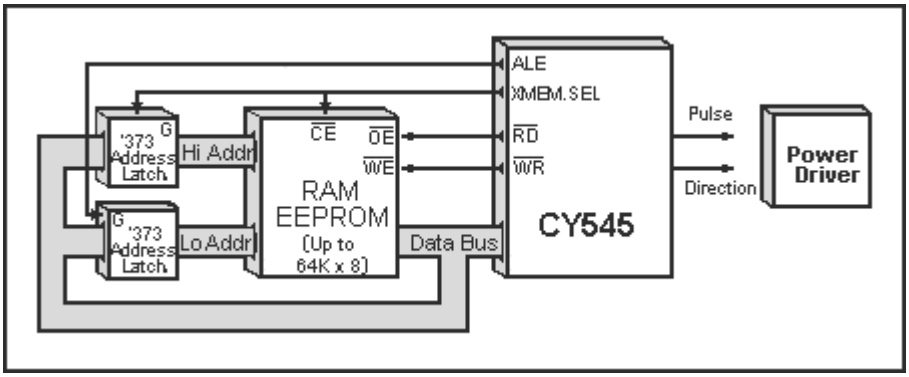
The CY545 may be operated with up to 64K bytes of external memory, which hold various sequences of commands to run. A number of CY545 commands are used to support the external memory, allowing you to set the memory address pointer, enter commands into memory, and execute commands from memory.

Several types of external memory may be used, including RAM, EEPROM, and EPROM. Note that EPROM memory must be programmed separately from its use in a CY545 application, so the CY545 Enter command and the ability to define the contents of memory are given up when an EPROM is used. The other two forms of memory allow the CY545 to write to memory as well as read it.

When RAM is used, the contents are lost when power is removed, unless some battery back-up power is provided. Normally, the contents of RAM must be redefined whenever the CY545 application is started.

With EEPROM, the memory can still be written by the CY545, especially if a 5-volt-only version is used, but the memory contents are not lost when power is removed.

An applications schematic, provided with this manual, illustrates the use of an EEPROM as the external data memory for the CY545. The device shown is a Xicor X28256 EEPROM, which has 32K bytes of memory. Smaller devices may be used if less memory is required, and a larger device, with a full 64K bytes of memory could be supported.

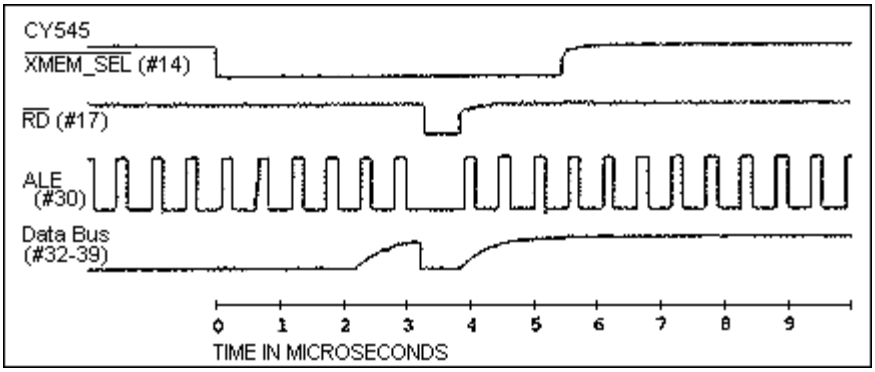


The CY545 provides all address and strobe timing signals to the memory. Signals involved are the data bus, XMEM_SEL, ALE, WR, and RD. Special use is made of the data bus, which provides all 16 bits of address, plus the actual data transfers. Two external latches are required to hold the address. The example uses 74LS373s for this purpose.

When the CY545 reads or writes the external memory, it starts by placing the most significant address byte on the data bus. It then selects the memory by driving XMEM_SEL low. This both chip-enables the memory and latches the upper address byte into the 74LS373 which holds address lines A8 to A15. The CY545 then performs a read or write operation.

The read or write is broken into two steps. First the lower address byte is generated on the data bus. This value is latched into the other 74LS373, using the ALE signal. Then the actual data transfer occurs. For a write, the CY545 generates the write data on the data bus, along with a write strobe, WR. For a read, the data bus is floated, and a read strobe is generated, using RD, so the memory can now drive the bus.

When the memory transfer is over, the XMEM_SEL signal goes high again, de-selecting the external memory. This allows the lower address byte and RD and WR strobes to be used with other hardware as well, without interference from the memory. Only the memory uses the upper address byte.



As a special feature, the CY545 performs a verify Read after writing to the memory. This function is provided in support of EEPROM memories, that require a long time to perform a write operation, and cannot perform another random write or read while the first one is in progress. Many of these memories support Data/ polling, in which at least one bit of data is inverted while the memory is busy writing the byte. When a read is performed, the data read does not match the data written until the write operation is complete. This change in data values indicates when the memory is ready for the next operation.

The CY545 performs the verify Read repeatedly until the data matches the value written, or a time out occurs. For RAMs, the match will occur immediately, so the CY545 simply continues at that point. For an EEPROM, the match will occur after the EEPROM write is finished, and the CY545 waits during this time before going on to the next function.

You should be aware of this time period delay when using EEPROM, because it is possible to send serial characters to the CY545 faster than the EEPROM can accept them, when running the serial interface at higher baud rates. It is best to have a time delay between each character in this case, or else use the CTS signal to hold off the communications until the CY545 is ready.

Note that the parallel interface takes care of this time delay on its own, since the CY545 will stay busy while it is writing the character to the memory.

"Auto-Start" Feature for Stand Alone Operation

Another special feature of the CY545 is the use of the external memory to automatically run a command sequence at power-up, or whenever the CY545 is reset.

When the CY545 is reset, it will read the first three locations of the external memory. If the values read are 12H, 34H, and 56H respectively, the CY545 will then switch to the program execution mode, and start reading commands from location 00003 of the memory. This allows the CY545 to work in a stand alone mode, without any commands from a host computer or other source than the memory itself. This function is shown below:

00000	12H	12H in address 0
00001	34H	34H in address 1
00002	56H	56H in address 2
00003	...	First CY545 command in address 3

This feature will only occur if the first three bytes have the proper key codes in them. Any other values in those locations will keep the CY545 in the direct command mode, waiting for commands from a host system before doing anything.

To set up the auto start function, simply record the data bytes 12h, 34h, and 56h, in locations 0 to 2 of the memory, followed immediately by the command sequence desired. For example:

Y 0<cr>	Set memory pointer address to zero
E<cr>	Following sequence is recorded to memory
<Ctrl-R>4V	key sequence 12h, 34h, 56h
R 125<cr>	standard CY545 commands follow now
F 15<cr>	
...	additional start up commands
Q<cr>	End of memory content definition

When the CY545 is now reset, it will execute the memory-based commands, starting with the “R125<cr>” command, since the key sequence was found in the first three locations.

As a special convenience, you may command the CY545 to ignore the auto start command sequence, by holding the XMEM_SEL signal low while the CY545 is reset. If the signal is low, the CY545 will not test the memory contents for the 12h, 34h, 56h key sequence. Instead, the CY545 will set itself into the Command mode, and wait for commands from the serial or parallel interface.

This is especially useful during program development, where errors in the memory content must be corrected before the CY545 can run the memory based commands. Without this feature, you would be forced to remove the memory and delete the key sequence, before the CY545 could be operated with the memory again.

The sequence for bypassing the Auto Start tests is:

- Drive the XMEM_SEL signal low
- Reset the CY545
- Delay 2 milliseconds after reset is released
- Release the XMEM_SEL signal
- Issue any CY545 commands in the direct Command mode

CHAPTER 13 - THUMBWHEEL SWITCH SUPPORT

Thumbwheel Switch Support

A very important feature for stand-alone control is the ability to define variable parameters for the control functions. In many applications, the basic function is similar for a whole set of operations, but the exact numbers will vary with the particular operation being selected. For example, a machine could be designed that takes raw stock, such as lumber, and cuts it to specified lengths.

As a CY545 program, such an application is fairly simple. If the stepper motor controls the feeding of the stock, you must simply move the motor a specific number of steps that correspond to the desired length, then activate a cutting mechanism. This function then repeats as long as stock is available, or for the number of final pieces desired.

If you now wish to cut stock of another length, the same command structure as before could be used, with only a change to the number of steps to move.

A number of options exist for implementing such a system. First, a host computer could be used to enter the desired function, and the computer program could generate the proper command sequence for the CY545. While this approach is quite practical, the host computer may be too expensive to justify for such a simple task.

Next, the external memory support of the CY545 could be used to hold several programs, one for each of the desired lengths needed by the application. Some mechanical selection process could then choose which program the CY545 would run, based on the desired length. This approach is also practical, so long as you can define all the required lengths within the memory contents. If a new length is desired, a program for that length must first be defined in the memory before it can be run. This requires the machine to be stopped, while a programmer enters the proper commands to the CY545.

Finally, it would be nice if only one program were required, since the basic function is always the same. This requires the CY545 to accept variable parameters, fixed at the time the program is run. This would allow any length stock to be selected, with a resolution of one step from one length to the next, without using a host computer or additional programs in external memory.

The CY545 provides such a function, using external thumbwheel switches to specify the parameter values. All examples shown so far have used fixed parameters as numeric arguments to the CY545 commands, such as Rate and Number. For example:

R 123<cr>
N 4321H<cr>

When such commands are read by the CY545, either from the command interfaces or from the external memory, the parameter value is fixed by the numeric argument, and is a constant for that command.

However, the CY545 will also accept another form of parameter for the arguments of the commands. **This form replaces the numeric argument with a "#" character**, and it means to read the actual parameter value from an external source, normally implemented as thumbwheel switches. Only the first parameter of a command with multiple parameters may be read this way, but this is not a severe restriction, since the second parameter is usually a branch address, and could be fixed anyway. Programs may repeat things a variable number of times, but usually go to the same place in the loop. Examples of this new format are shown below:

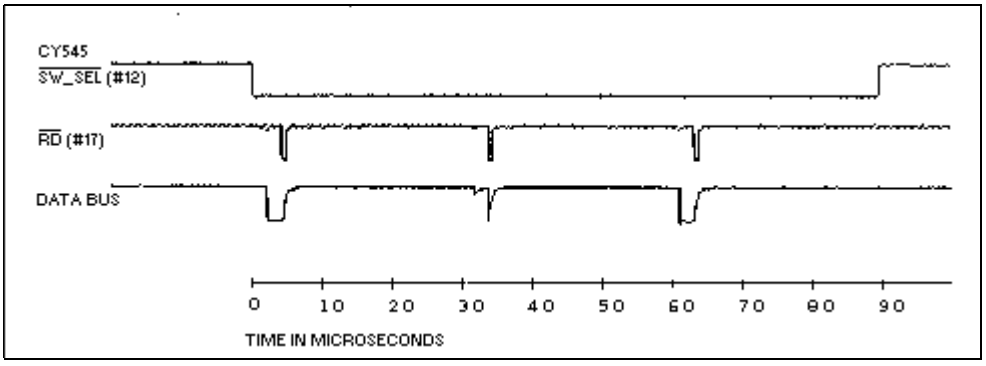
R #<cr>	Get Rate value from thumbwheels
N #<cr>	Get Number of steps from thumbwheels
L #,123<cr>	Get Loop repeat count from thumbwheels

In all cases, the format for the command is identical to that with the numeric arguments, except that the numbers are replaced by the "#" character. Note that this format only works in the ASCII command mode, since the "#" character cannot be distinguished from a valid parameter byte value in the Binary command mode. All other characters of the command stay intact, including the space after the command letter, comma or space between arguments, and carriage return at the end of the command.

When these commands are written to external memory, the "#" character is included in the memory contents. Recall that the external memory contents are identical to the characters sent in the direct command mode. The CY545 will act on the character, and read the parameter value every time the program is run, and a command containing the "#" character is read.

When the "#" character is processed by the CY545, it generates a special read sequence to get the value of the parameter from the external switches. Each parameter consists of 3, 5 or 8 digits, depending on whether the parameter is a byte or two byte size. Each parameter is assigned an address range of 3, 5, or 8 addresses, generated on the lower byte of the two address latches, controlled by the ALE signal.

In order to read a specific digit, the CY545 generates the digit address, latched into the lower address byte latch. It then generates a RD strobe, with the SW_SEL line low. This should enable the data from the addressed switch onto the CY545 data bus. All switches are positive BCD encoded. The CY545 reads the switch value from the lower four bits of the data bus, then converts the value into an ASCII decimal digit. The hex format is not supported for externally read parameter values.



To read an entire parameter, the digit read process is repeated three, five, or eight times, depending on the size of the parameter. The CY545 starts by reading the most significant digit, then the next, until it finally reads the least significant digit. This is the same order that digits are read from a command with numeric arguments.

The various CY545 commands have been split into three groups for parameter addressing purposes. This organization minimizes the address decoding logic needed to access the external switches.

The applications schematic includes an example interface to three different parameters, the Number parameter with 5 digits, and the Rate and First rate parameters, each with 3 digits.

Each group of 8 parameters is selected by a 74LS138 decoder, which generates an enable signal for the selected command parameter. Each digit of the enabled parameter is uniquely enabled by a signal from a 74LS156 decoder. Digits are enabled in address sequence, with the least significant digit selected by the Y0 output selection from the 74LS156.

For 5 and 8 digit parameters, one 74LS156 must be used to address the parameter. It can generate up to 8 enable signals, but 3 will not be used in the 5 digit cases. The Delay parameter would be an example of this.

For 3 digit parameters, one 74LS156 can be used to address two different parameters. Each half of the 74LS156 can select 3 digits, with one line not used. The Rate and First rate parameters illustrate this example.

To allow every parameter to be specified by a thumbwheel switch selection would require three 74LS138s, plus one or one half 74LS156 per parameter. It also would take one thumbwheel switch per parameter digit. This could be a significant amount of hardware if all parameters are externally controlled. However, we expect that in real applications, only a few parameters will need this feature, and hardware complexity will not be that bad.

The thumbwheel switches used for each digit should be BCD encoded, and must have isolation diodes between the digit lines. Some switches have these diodes built in, but most do not. The diodes have been shown on the example schematic. Also, with the parts shown, the switches must have a complemented common signal to generate the proper values at the CY545 data bus. By using an inverting buffer to the CY545 data bus, a non-complemented common signal switch could also be used. Pull-up resistors provide a high signal for those digit lines not connected to the common for a particular digit value.

The following section gives the address values for each parameter the CY545 can read from external switches. The lowest number of the range corresponds to the least significant digit of the parameter, and is read last by the CY545. The highest number is the most significant digit address, and is read first.

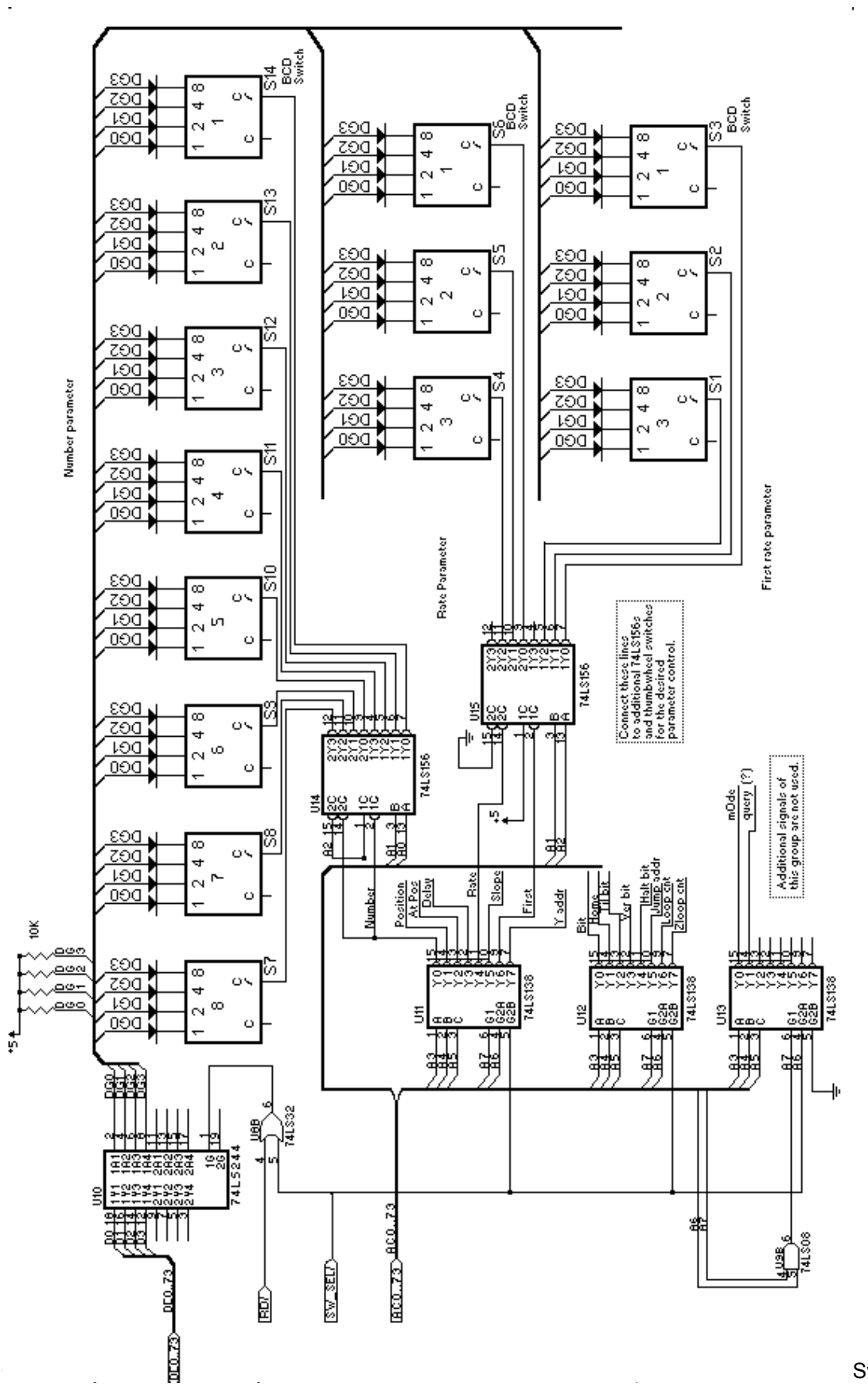
CY545 Thumbwheel Switch Parameter Addresses

Addresses	Parameter	
Group 1		
87H - 80H	**N** command number of steps	8 digits
8FH - 88H	**P** command target position	8 digits
97H - 90H	**A** command at position	8 digits
9CH - 98H	**D** command delay value	5 digits
A2H - A0H	**R** command step rate	3 digits
AAH - A8H	**S** command acceleration slope	3 digits
B2H - B0H	**F** command first rate	3 digits
BCH - B8H	**Y** command memory address	5 digits
Group 2		
42H - 40H	**B** command bit number	3 digits
4AH - 48H	**H** command bit number	3 digits
52H - 50H	**T** command bit number	3 digits
62H - 60H	**W** command bit number	3 digits
6AH - 68H	**J** command byte address	3 digits
72H - 70H	**L** command repeat count	3 digits
7CH - 78H	**Z** command repeat count	5 digits
Group 3		
C2H - C0H	**O** command mode value	3 digits
CAH - C8H	**?** command query letter code	3 digits

Each parameter has been assigned a potential range of 8 addresses, to make the address decoding easier. Only the addresses shown are actually used to read digit values. The CY545 will not generate the unused address values.

Any parameter that will not be read from switches does not require the 74LS156 or switches for that parameter. Any group of parameters that will not be read from switches does not require the 74LS138 decoder either.

CY545 Thumbwheel Circuit Diagram



CHAPTER 14 - CY545 OUTPUT DISPLAY SUPPORT

Output Format

As mentioned in the command descriptions, the CY545 can support a number of output devices, selected by the Mode command. Two types of serial output are possible, and two types of parallel output are possible.

The CY545 generates output from the string function, for user encoded messages, and from the Query command, for current parameter values. The exact format of the output is determined by the selected display and the ASCII or Binary command mode.

For standard serial output, the power-on default, the CY545 simply transmits each character of a string or query on the serial output line, TxD. Every character between the quotes is sent for a string, and the query parameter outputs are:

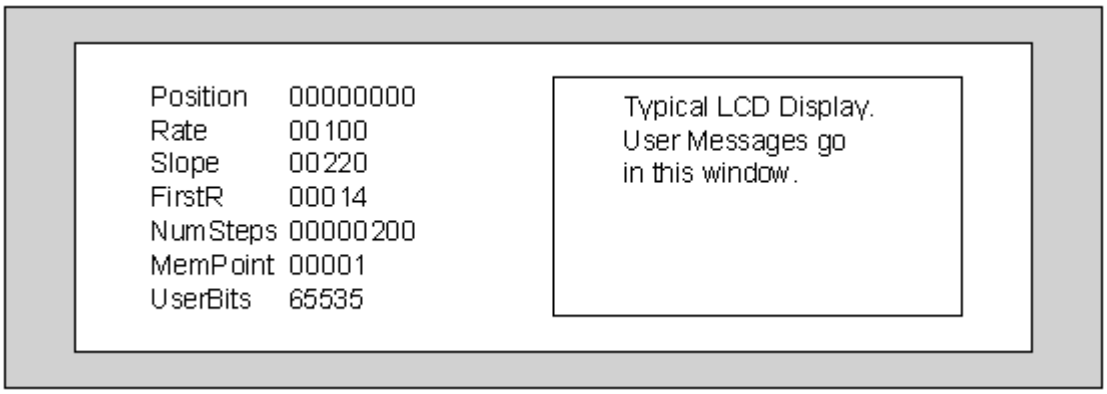
X=vvvvv<cr> or **X=vvvvvvvv<cr>**

Where "X" is the parameter letter (N, P, R, etc.), and "vvvvv" is the five decimal digit or eight decimal digit value of the parameter. The query output ends with a single carriage return character.

Output to LCD (CY325-Based)

When the CY325 based serial display is selected, the CY545 commands the CY325 to split the display into two sections. The right hand section is used for string outputs, and no modifications are made to the characters of a string. The left hand section of the display is used for the parameter values, with each parameter assigned a different line on the display.

When a parameter query command is received, the output of the value involves moving the display cursor to the proper display location, then printing the parameter value, and finally, moving the cursor back to the standard display area.



The CY325 is a special display control chip available from Cybernetic Micro Systems.

Output to Parallel Device
(Including CY233 Network Chip)

The standard parallel output can be used by a parallel connected host or by the CY233 Network Communications Controller. Both connections use the same signals, the parallel handshake control lines IO_REQUEST, BUSY, and FPL (USRB6).

When the CY545 receives a Query command or a string to send, it changes to the output mode internally, and lowers the FPL signal. It then waits for a low level on the IO_REQUEST signal, indicating that the host or CY233 is ready to take a character.

The character to output is placed on the data bus, and the CY545 drives the BUSY line low, indicating that a character is available.

The CY545 will now wait in this state until the host or CY233 drives IO_REQUEST high again. This indicates that the character has been accepted, so the CY545 removes the character from the data bus, and raises the BUSY line again.

This handshake process repeats until all the characters of the Query output or string have been sent. The CY545 then drives the FPL line high again, and goes back into the command mode.

A CY233 interface is included with the “CY545 Applications Example”, shown in a previous chapter.

Output to 8-Digit LED Display

The final selection for a display device is a special parallel display, using the Hewlett-Packard HDSP-211x displays. These displays are very compact, LED-based, 8 character, dot matrix displays. They have programmable intensity and custom definable character fonts.

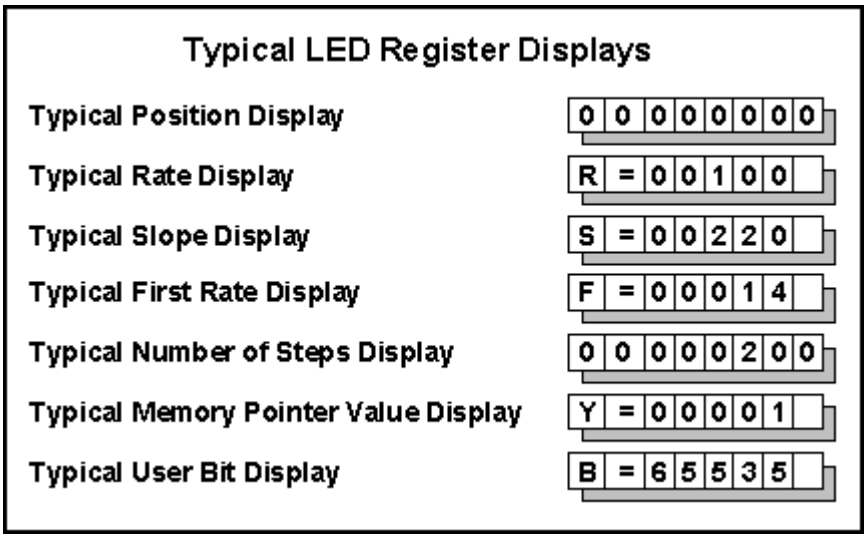
The displays connect to the CY545 over the parallel data bus, the lower address byte, and the HP_SEL (USRB7) signal. The RD and WR strobes also provide data transfer timing for these displays.

The applications example schematic includes the proper connections for the HDSP-211x displays. This connection scheme must be used for proper operation of the displays, because the CY545 assumes the address lines are connected as shown to access the display control registers and data buffers.

When the HP display is selected, any string messages are sent to the display verbatim. Be careful that messages are not longer than 8 characters at a time. Carriage return codes are not displayed, but they reset the character address to the left most display character, and cause the display to be blanked before the next character is written.

Query command outputs follow the same format as the standard serial and parallel display outputs, except for the 8 digit values, which only display the value, so the HP displays can show one parameter value at a time. The display is blanked, and a new parameter value is written each time the Query command is used.

The CY545 takes care of any timing involved in running the HP display, and generates the proper control signals. Also, a special command ([ ) allows you to customize the display control for your desired intensity or other functions.



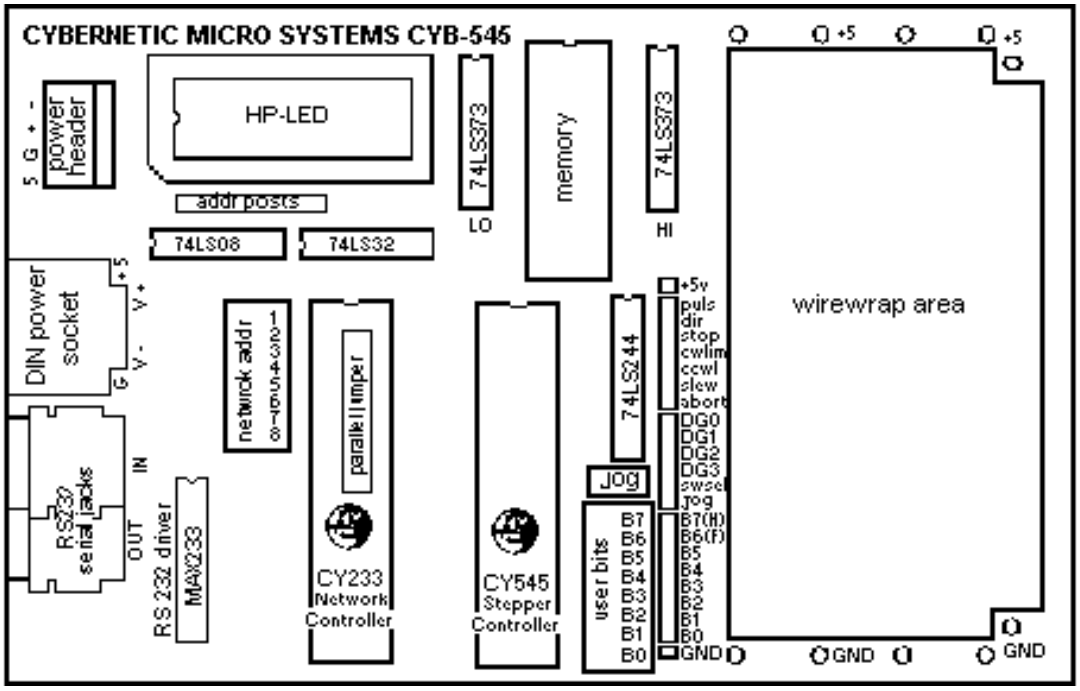
CHAPTER 15 - CY545 PROTOTYPING SUPPORT

The CYB-545 Prototyping Board

The CYB-545 (and later CYB-550) Prototyping Board implements the Example Circuit - Maximum configuration as discussed earlier, supporting all features of the CY545. The CY545 may communicate serially with the host over an RS232 interface using the RJ11 jacks, while parallel communications are supported on a 20-pin header.

Multiple axes or networked controllers are supported using the CY233 Network Controller. The CY233 communicates with the CY545 over the Parallel bus.

This board also contains the interface circuits to support an HP or Siemens 8-character LED, for displaying parameter values, prompts and messages. Up to 256K bits (32Kx8) of memory are addressable at the memory socket, which can accept PROM, EPROM or EEPROM. It is possible for such a system to operate without the use of a host computer. The CY545 with local memory can contain the routines required to run the system, while the display provides the mechanism for system control by the operator.

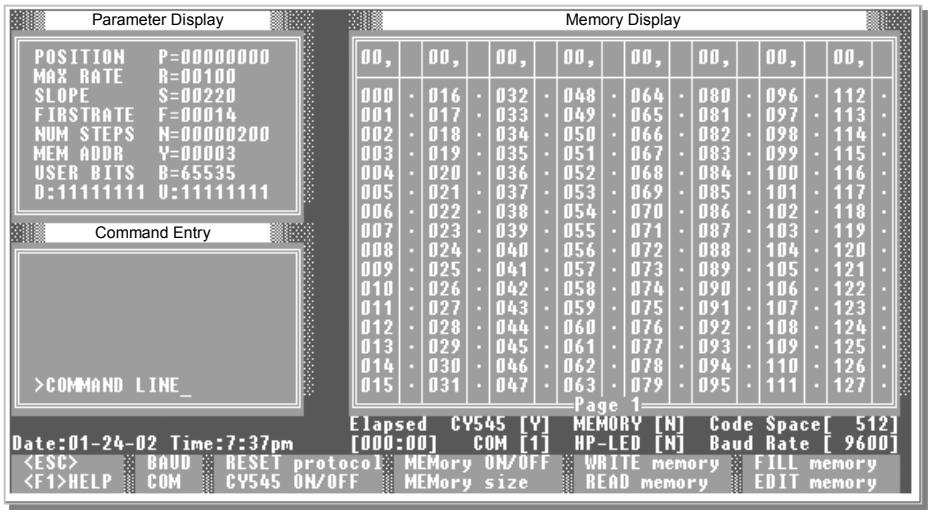


A Wirewrap area is provided for implementation of thumbwheel switches, driver circuits, or other custom circuitry. Based in the single height VME format (100mm x 160mm, approximately 4" x 6.3"), the board requires only +5v (@650mA). The RS232 level voltages are generated by the Max233 on the board.

The basic board is provided in kit form, with all parts necessary to assemble a working board. Separate options are available, including an LED Display, Memory, EEPROM, CY233 Network Controller Chip, power supply, and serial cable.

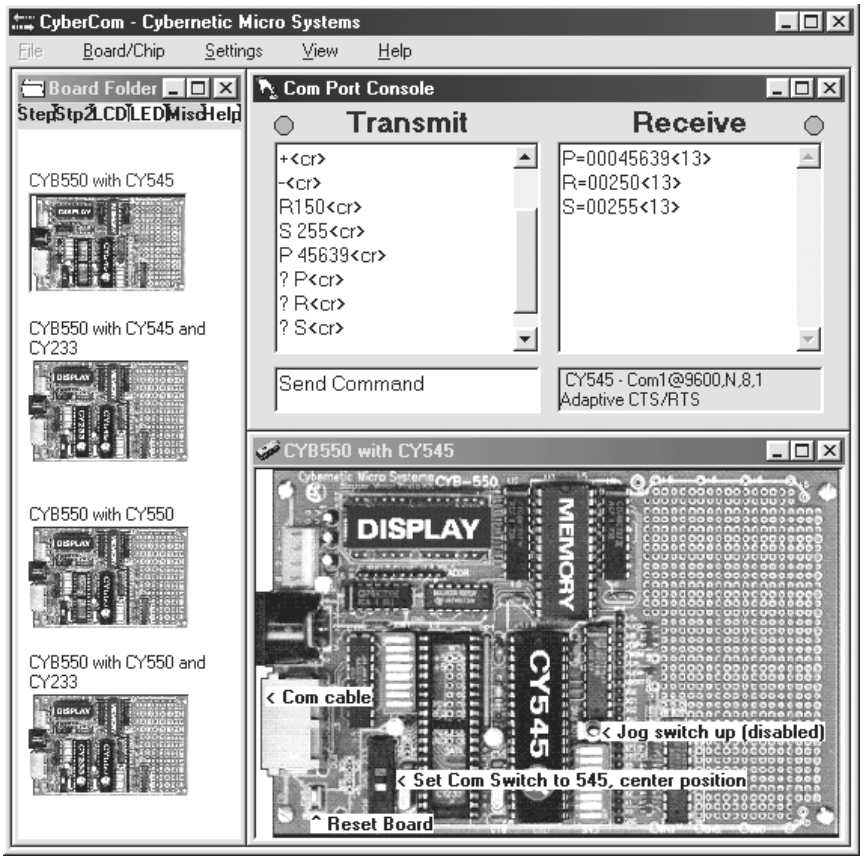
CYS-545 Panel Software

The Dos-based CYS-545 uses an IBM-PC Com port directly communicate with the CYB-545. Enter, change, or read back stepping parameters; write and save external programs to EEPROM; or read and edit the externally stored program. As a learning tool, the software filters each command for proper format and appropriateness, and the help screens include command summaries and a pinout diagram.



CyberCom Console Software

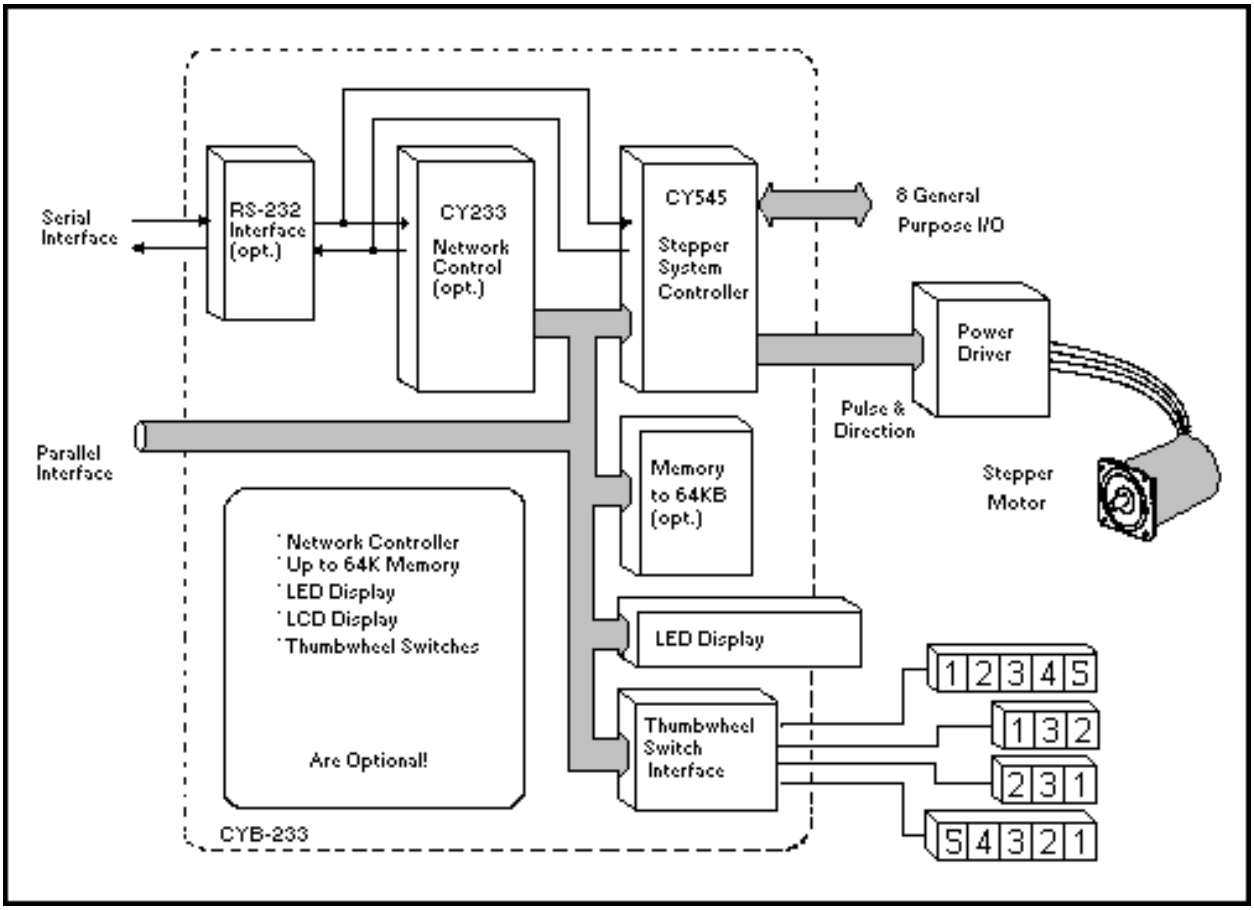
This Win98/NT-based software Provides Com Port Communication for Cybernetic Micro Systems' Controller Chips and Prototyping Boards. The program allows keyboard entry to one of Com 1, 2, 3, or 4. The user has control over many of the Com port settings, including Baud, data-length, parity, stopbits, and CTS/RTS. Typing is buffered at the keyboard until the Enter key sends the entire string to the serial port. A Carriage return terminator is optional.



This utility directly supports all of the Cybernetic prototyping boards that have RS232 connectors. Default settings are provided for each board, and all the boards are pictorially depicted with proper jumper and switch settings and startup tips.

The CYB-233 Prototyping Board

One general purpose prototyping board available from Cybernetic Micro Systems is the CYB-233 board. While this board was originally designed to support the CY233 Local Intelligent Network Controller, it also contains a target socket area that will accept a CY545.



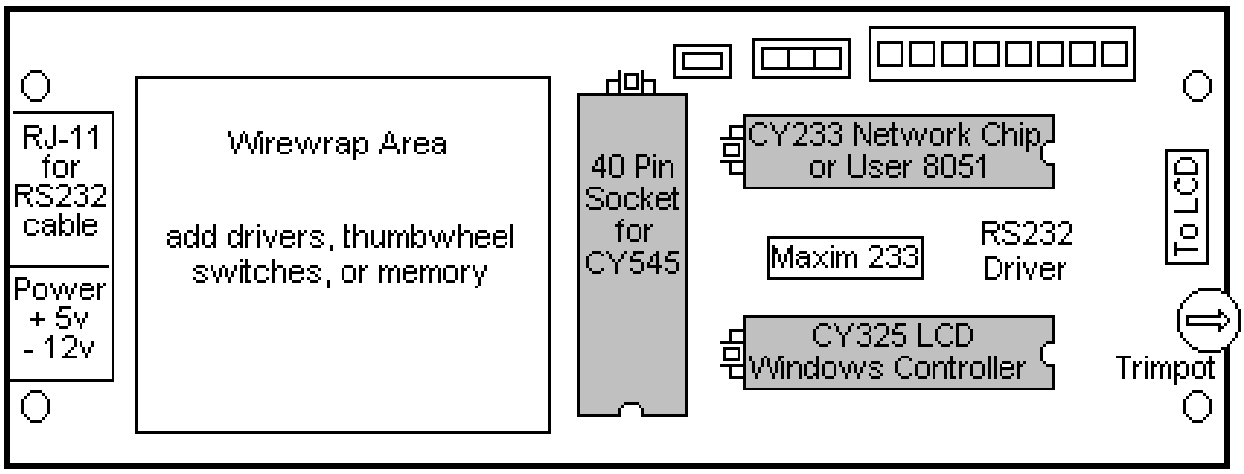
The CY545 can be commanded directly through its own serial port connection, or serially through the CY233 and the parallel CY545 interface. With this second option, it is possible to control multiple CY545s through one host serial port, as a network of motor controllers. The CY233 acts as the network communications controller, recognizing commands meant for the local CY545, while the CY545 runs the stepper motor as commanded by the host.

The CYB-233 board also contains a wire wrap area, to which you may add any special motor driver interface logic, the local external memory, thumb wheel switch interface, or HP LED display. This makes the board versatile enough to accommodate most CY545 applications, at least for prototyping purposes.

The board is supplied in kit form, with complete assembly instructions and basic components.

The CYB-003 Prototyping Board

Those CY545 applications that use the CY325 and liquid crystal display, are well supported by a second prototyping board from Cybernetic Micro Systems, the CYB-003 board. This board contains the interface circuits and connections to support the CY325 and display unit. The board is the same size as the popular 711 LCD display, which is 8 lines by 40 characters.



The board contains a socket that will accept the CY545, and allows it to communicate serially with the CY325. Also included are an RS-232 interface and RJ-11 jacks for serial connection to the board. A wire wrap area makes it possible to add the motor driver interface, the local external memory, and thumbwheel switch interface for the CY545.

The combination of the CYB-003 board, with the CY545, CY325, and any interface logic, plus the 711 display, makes a very compact stepper controller system. This is especially useful in stand alone applications, where the local memory holds the CY545 stepping routines, and the CY325 plus display are used for the operator interface, status display, and prompting messages.

CHAPTER 16 - TIMING AND CONTROL

Timing and Control

This section contains more detailed timing and applications information for various signal groups of the CY545. Included are the command interfaces, special external hardware support, and motor control signals.

The Parallel Handshake

The CY545 parallel command interface uses three control signals, plus the data bus lines for actual data transfers. The control signals are IO_REQUEST, BUSY, and FPL, which is a shared function with USRB6. IO_REQUEST is an input, driven by the device sending commands to the CY545, while BUSY and FPL are outputs, indicating status of the CY545.

The IO_REQUEST signal is used to control the parallel handshake functions, The signal is normally high when no commands are being sent to the CY545. To issue commands, wait until the BUSY line is high, indicating that the CY545 is ready for commands, then put a command byte on the data bus lines, and drive IO_REQUEST low. Keep the data signals and IO_REQUEST stable until the CY545 responds by driving BUSY low. Note the response time is normally around 25 microseconds (usec), so the CY545 is slower than an I/O peripheral or memory device, and some systems may require a latch on the data bus signals. Once the BUSY signal has gone low, the IO_REQUEST signal should be driven high again, and the data signals may be removed. A typical handshake waveform is shown below.

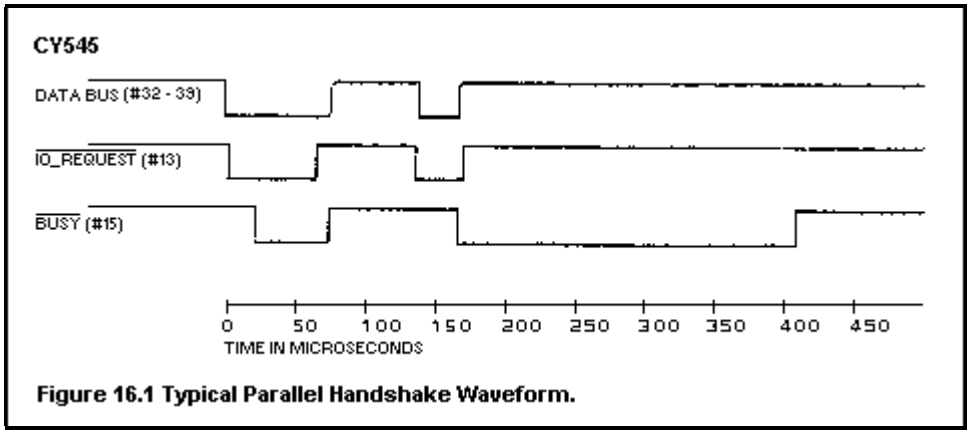


Figure 16.1 Typical Parallel Handshake Waveform.

Every command character must be sent to the CY545 with the above described handshake. Timing for the handshake will vary somewhat, and the CY545 will go busy for longer periods of time after the last character of a command is issued. This represents the actual execution time of the command.

When the command is a query command, the CY545 will generate a response to the query as part of the command execution. If the mode register is set so that query responses come out the standard parallel interface (MB0 = 1 and MB2 = 0), a handshake will also be used for the data generated by the CY545. In this case, the CY545 will drive the FPL signal low to indicate that it

wants to generate some output to the parallel interface. The CY545 then waits for the IO_REQUEST signal to go low, indicating that the host system is ready for data. The data byte is written to the data bus, and the BUSY signal is driven low. This indicates that data is available to the host. When the host drives the IO_REQUEST signal high again, the bus is cleared, and BUSY is brought back high also.

In the ASCII command mode, the FPL signal will continue to stay low as long as the CY545 has data to generate. This is an indication that the host should continue reading data bytes from the CY545. The FPL signal will be brought high when the CY545 writes a carriage return code, indicating the end of the response.

In the Binary command mode, data may have any value, so each byte sent out will toggle the FPL line back high again. When there are multiple bytes to send, the FPL signal will be brought low as the CY545 is ready to generate the next data byte value.

This behavior is compatible with the CY233 LINC device, generating one message for query responses in the ASCII mode, and one message per data byte in the Binary mode. A typical query response waveform is shown below.

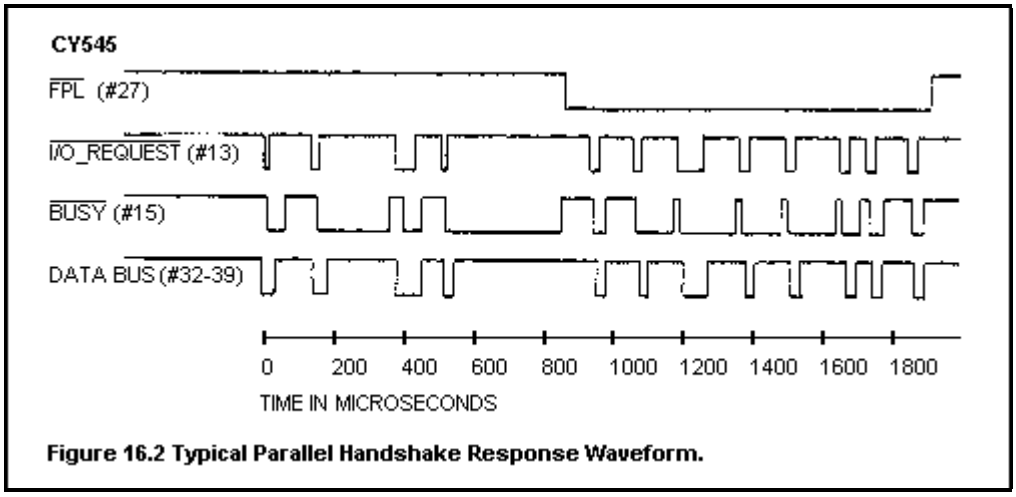


Figure 16.2 Typical Parallel Handshake Response Waveform.

Immediate Commands During External Memory Program Execution

The CY545B supports the acceptance and proper execution of immediate commands while executing a program from external memory. However, the handshakes for this function are a little more complex than the standard command handshake. More details are provided in this section.

The FPL signal is very important in implementing the handshake protocol for immediate commands, especially if the CY545 is also generating query responses or quoted messages from the local memory program.

The FPL signal, driven by the CY545, can be used to control the direction of data on the CY545 local data bus. When FPL is high, the CY545 can accept command inputs, and when FPL is low, the CY545 has output data for the parallel host.

In the CY545B, the FPL signal is driven low when the CY545B has data to output, and it is removed after the host has read the last character, and brought I/O Request high again, but before the CY545B brings Busy high. This timing is compatible with hardware implemented handshakes, requiring no time delays or complex logic to implement.

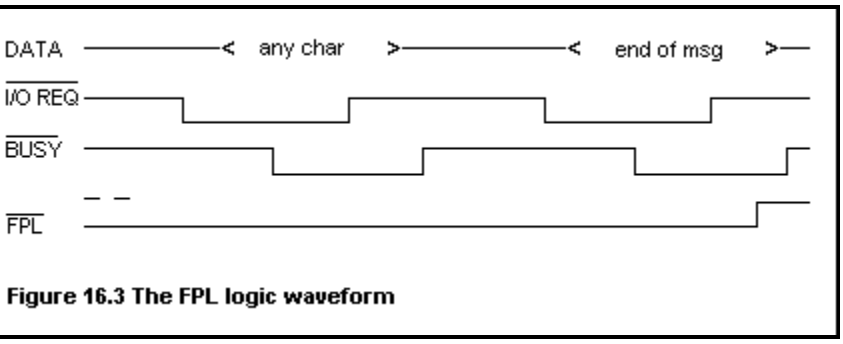


Figure 16.3 The FPL logic waveform

Notice that the FPL signal is removed at the end of the message, while the CY545B is still Busy, but after the I/O Request has been driven high again.

Also, there is a fundamental handshake change required for the CY545B to accept immediate commands from the host, while reading and executing commands from the local external memory.

If the CY545B is to execute a program from the local external memory, and accept immediate parallel commands at the same time, there is a resource conflict involving the data bus. This bus is used to both read commands from the memory, and read commands from the parallel host, while only one device is allowed to drive the bus at a time. Notice that the CY545B may also drive the bus while outputting messages to the host or the local HP display! This makes the data bus a very critical resource in CY545B designs.

In order to allow immediate commands to be issued at any time, even while the CY545B is reading data from the local memory, we must implement a new handshake protocol for the immediate commands, which prevents corruption of data on the data bus. In the CY545B, this is done without using any more signals, so no other User Bits are given up to perform this more complex handshake.

Instead of using additional signals, immediate commands are isssued with a modified first handshake, involving no character! That is, the host drives I/O Request low, and waits for the Busy signal to go low, but the data bus is not driven. This allows the CY545B to finish reading and executing any command, from the local external memory, that is in progress when the host decides to issue the immediate command.

After the memory command is finished, including possible output of a message or query response, the CY545B will drive Busy low, with FPL high, to indicate that it is now accepting the host's immediate command. This is similar to a normal command handshake, except that the command letter is not put on the data bus.

Since the data bus is not driven during this first handshake, the CY545B reads a 0FFh data byte (due to the pull-up resistors on the local data bus), and will ignore this byte.

This special, first character handshake is followed by the normal command format, starting with the normal command letter, which the CY545B reads with a normal handshake. After executing

the immediate command, the CY545B resumes external memory execution, unless the immediate command was the "0" command, which stops the execution.

Note that the first handshake, without driving the data bus, is only required by the parallel interface during immediate commands, issued while the CY545B is also running a program from the local external memory.

When the CY545B is only executing immediate commands from the host, or when the host uses the serial interface instead of parallel interface, the "0FFh" first character is not required. However, if you issue a first handshake with no command character (read as 0FFh by the CY545B) in these cases, the CY545B will still ignore that character.

The examples below show the initial handshake of an immediate command during external memory execution. Notice that in the general case, FPL might go low before Busy goes low. In this case, the CY545B has decided (due to a command from the memory) to output a message to the host rather than accept the immediate command input. This is shown in the first example waveform below. The CY545B is driving the data bus when FPL is low.

The host must accept the output message from the CY545B before continuing with the immediate command in this case. The state of FPL during the first handshake will determine if the CY545B is taking the command from the host (FPL high), or if the CY545B is outputting a message to the host (FPL low).

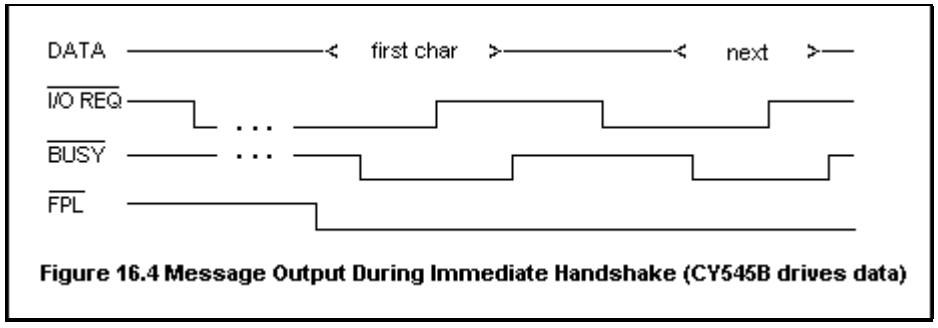


Figure 16.4 Message Output During Immediate Handshake (CY545B drives data)

After the CY545B has issued the output message, and FPL is high again, the host may continue with the immediate command handshake, and the CY545B will now accept and execute the immediate command as shown in the second example waveform below. Here, the host does not drive the data bus for the first handshake, and thereby gives the CY545B the "0FFh" first command character prefix. This is followed by the normal first command character.

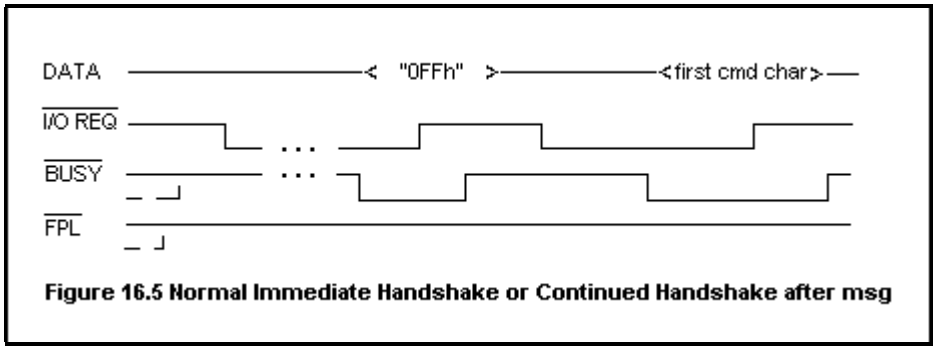


Figure 16.5 Normal Immediate Handshake or Continued Handshake after msg

Also note that the second example waveform applies to the case where the CY545B accepts the immediate command without first generating some output data.

In addition, the CY545B will only issue an output message and take control of FPL if there is a query or quoted message in the memory program. If there is no such command, the CY545B will always accept the immediate command from the host, with no chance of changing FPL to the output mode during the first handshake of the immediate command.

In this case, the host handshake is less complex, because the host is not required to test the state of FPL during the first handshake with no data. FPL should always be high. However, during the first handshake the data bus must still be left floating, so the CY545B can continue to read from memory, if required, to finish the currently executing command.

When using the serial command interface, it is also possible to issue immediate commands to the CY545B while it is executing a program from the local external memory. As with the immediate parallel commands, the CY545B will stop the memory execution, process the immediate command, then continue execution from the memory, unless the immediate command was the "0" command.

Since the CY545B may be in the middle of a command when the host decides to send the immediate serial command, it is best to use the Clear To Send ("CTS") protocol to help control this environment. This will prevent overflow of the CY545B serial buffer, since it cannot read the new command in the middle of executing the command read from memory.

The CTS signal is enabled by setting MB5 of the mode register to one, using the Mode ("O") command.

The CY545B will drive CTS high (off), if enabled, during the execution of any memory command, then drive it low again as the next memory command is being read. If the host uses the CTS signal to indicate when it is safe to send a command letter, the CY545B will accept the letter, and save it in the serial buffer until the current memory-based command is finished. It will then read the immediate serial command letter, and wait for more serial data, to complete the command. CTS will stay low during this time.

Once the immediate command has been received, the CY545B will execute it, with CTS driven high. External program execution will then resume after the immediate command.

As with the parallel command interface, the CY545B does not mix the command read from memory with characters received over the serial interface. Once the first letter of a serial command has been received, execution from the external memory is suspended until the immediate command is finished.

The Serial Interface

The CY545 serial command interface uses two primary signals, RxD and TxD. Serial data is received on RxD and is sent on TxD. Note that when the parallel interface is not used, IO_REQUEST and BUSY may be tied to select a fixed baud rate for the CY545. Otherwise, the adaptive mode will be used, or a parallel mode command can select a fixed baud rate (either from the local external memory, or from a parallel-connected command source). In the adaptive case, the CY545 must receive two carriage return codes before receiving or sending any other serial data. It will adjust the baud to match that of the received carriage return codes. With an 11 MHz crystal, the CY545 can adapt to standard baud rates between 300 and 19,200 baud.

The CY545 optionally implements a third serial control signal, the CTS signal. You must issue a Mode command to the CY545 to enable this signal, which shares functions on USRB6. With this signal enabled, the CY545 indicates to the host system when it is ready for more serial data. The CTS signal will be turned off when the CY545 is busy executing a command. For long duration commands, such as time delays or motions, this will inhibit the host from sending more commands until the CY545 is ready for them, providing an automatic serial handshake.

The CY545 has a three character serial buffer, which should be enough to let the host system respond to the CTS signal. Otherwise it does not buffer serial received data. The CY545 can accept serial data continuously within the range of the standard baud rates, until the end of a command is received (carriage return in ASCII mode). It will then be busy processing the command, and may take some time, if the command involves a motion, time delay, wait for external signal, etc. During this time, the CY545 cannot accept any more serial data, and none should be sent by the host. If the CTS signal has been enabled, this signal will indicate that no more data should be sent. Otherwise, the host must perform a time delay, allowing the CY545 to execute the current command, before sending more command characters.

When a query command is received, and the CY545 mode register is set to respond out the serial port (Mode bit MB0 = 0), the query response will be sent on the TxD signal after the command is received. Note it is possible to issue the query command on the parallel interface, and have the response sent out the serial side. A typical serial query and response are shown below.

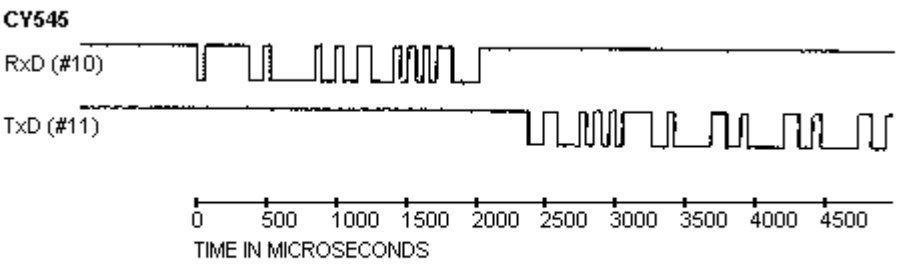


Figure 16.6 Typical Serial Interface Waveform

External Memory Control Signals

As discussed previously, the CY545 can support up to 64K bytes of external data memory, used for storing and executing various command sequences.

The CY545 provides all address and strobe timing signals to the memory. Signals required are XMEM_SEL, ALE, WR, RD, and the data bus lines. Special use is made of the data bus, which provides all 16 bits of address, plus the actual data transfers. Two external latches are required to hold the address. The example shown uses 74LS373s for this purpose.

When the CY545 reads or writes the external memory, it starts by placing the most significant address byte on the data bus. It then selects the memory by driving XMEM_SEL low. This both chip-enables the memory and latches the upper address byte into the 74LS373 which holds address lines A8 to A15. The CY545 then performs a read or write operation.

The read or write is broken into two steps. First the lower address byte is generated on the data bus. This value is latched into the other 74LS373, using the ALE signal. Then the actual data transfer occurs. For a write, the CY545 generates the write data on the data bus, along with a write strobe, WR. For a read, the data bus is floated, and a read strobe is generated, using RD, so the memory can now drive the bus.

When the memory transfer is over, the XMEM_SEL signal goes high again, de-selecting the external memory. This allows the lower address byte and RD and WR strobes to be used with other hardware as well, without interference from the memory. Only the memory uses the upper address byte. A typical memory access waveform is shown below.

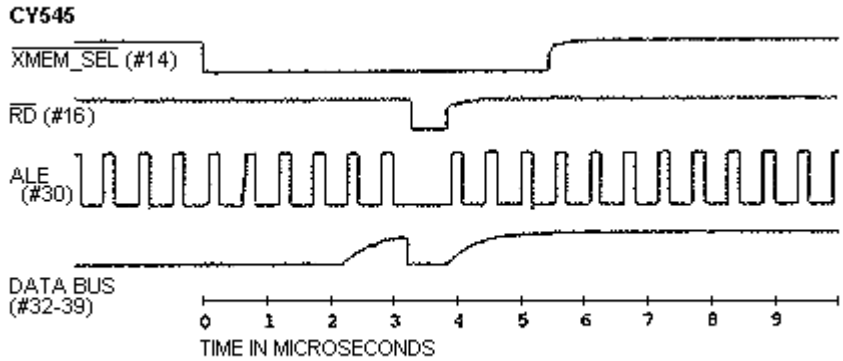


Figure 16.7 Typical External Memory Access Waveform

As a special feature, the CY545 performs a verify read after writing to the memory. This function is provided in support of EEPROM memories, that require a long time to perform a write operation, and cannot perform another random write or read while the first one is in progress. Many of these memories support Data/ polling, in which at least one bit of data is inverted while the memory is busy writing the byte. When a read is performed, the data read does not match the data written until the write operation is complete. This change in data values indicates when the memory is ready for the next operation.

The CY545 performs the verify read repeatedly until the data matches the value written, or a time out occurs. For RAMs, the match will occur immediately, so the CY545 simply continues at that point. For an EEPROM, the match will occur after the EEPROM write is finished, and the CY545 waits during this time before going on to the next function. This feature is shown in the waveforms below.

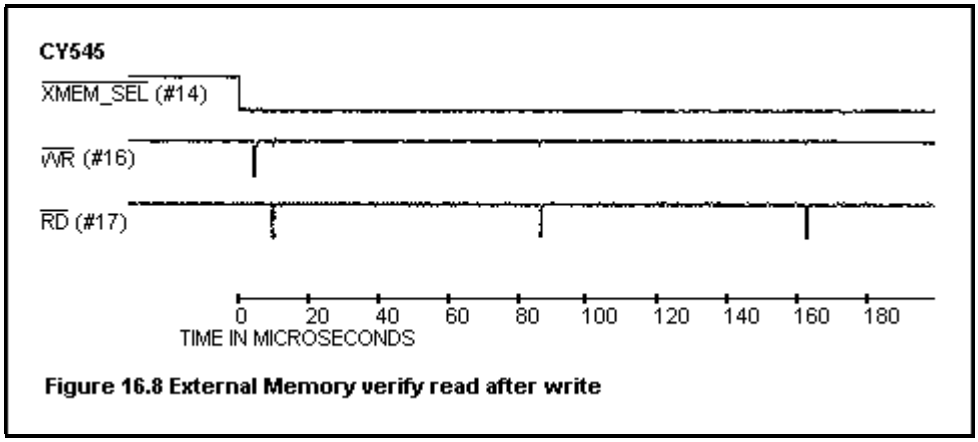


Figure 16.8 External Memory verify read after write

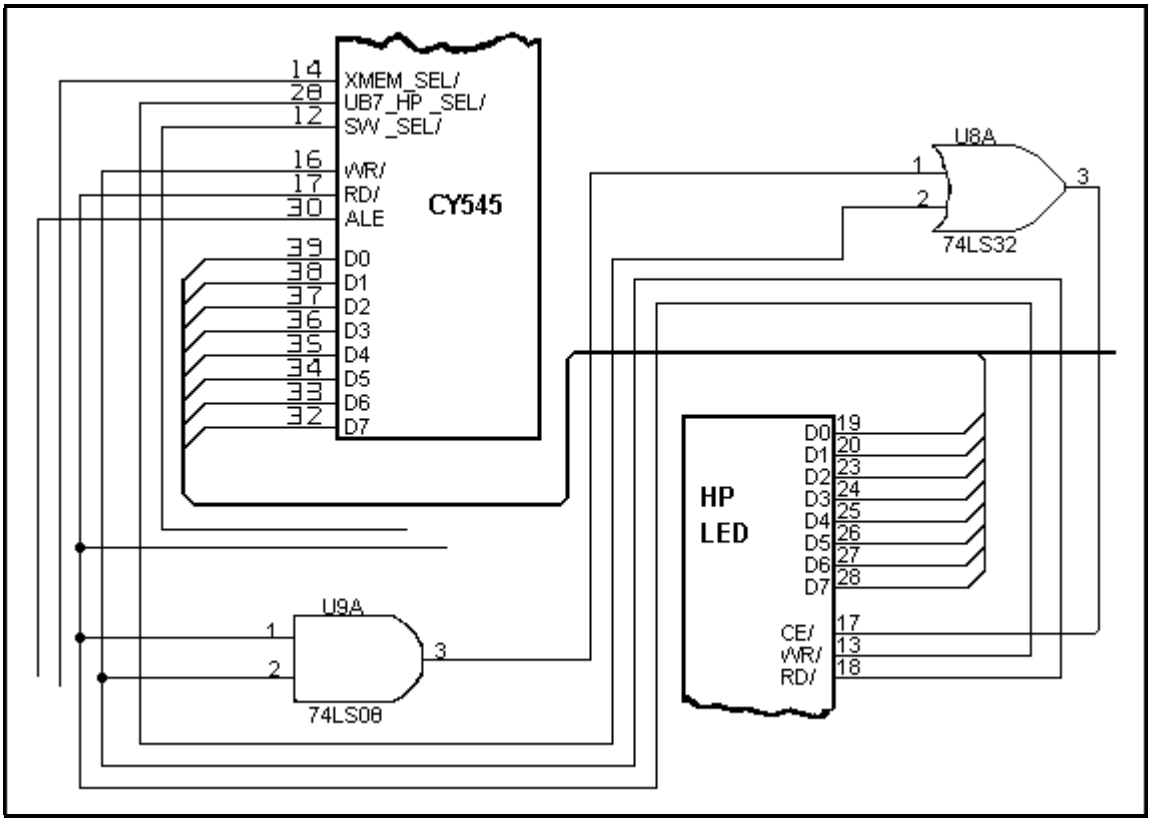
You should be aware of this time period delay when using EEPROM, because it is possible to send serial characters to the CY545 faster than the EEPROM can accept them, when running the serial interface at higher baud rates. It is best to have a time delay between each character in this case, or else use the CTS signal to hold off the communications until the CY545 is ready.

Note that the parallel interface takes care of this time delay on its own, since the CY545 will stay busy while it is writing the character to the memory.

Additional care must be taken when using the parallel command interface along with external memory. Since both functions share the CY545 data bus, the parallel command source must be able to release the data bus when the CY545 is commanded to access the external memory. The host system should only drive the data lines when the CY545 is not busy, and the host must stay off the bus while the CY545 is executing commands from the memory. No special signals are provided in this mode; the CY545 simply reads commands from the memory and executes them until it encounters a stop command.

Special HP LED Display Control Signals

Along with the normal serial and parallel data outputs, the CY545 also supports a special parallel LED display, using the Hewlett-Packard HDSP-211x displays. These are compact, 8-character displays, with programmable intensity, custom characters, and other features. The displays are very convenient and compact, making them useful in a minimal CY545 based system.



The CY545 provides all address and strobe timing signals to the display. Signals required are HP_SEL (shared function with USRB7), ALE, WR, RD, and the data bus lines. For the HP display, only the lower 6 address signals are used, and they are generated from the data bus and ALE signals. The address signals must be demultiplexed by an external latch, such as the 74LS373. If external memory is also being used in the system, the same address latch used for the lower byte of the memory address may be connected to the display. This is shown in the example schematic.

The HP _SEL, RD, and WR signals must be combined as shown to generate the Chip Enable signal for the display. This signal has special timing requirements, and may not be simply connected to the HP_SEL signal.

When the CY545 reads or writes to the display, it starts by driving HP_SEL low. The read or write is then broken into two steps. First the lower address byte is generated on the data bus. This value is latched into the 74LS373, using the ALE signal. Then the actual data transfer occurs. For a write, the CY545 generates the write data on the data bus, along with a write strobe, WR. For a read, the data bus is floated, and a read strobe is generated, using RD, so the memory can now drive the bus.

When the data transfer is over, the HP_SEL signal goes high again, disabling the RD and WR strobes for the display. A typical display access waveform follows.

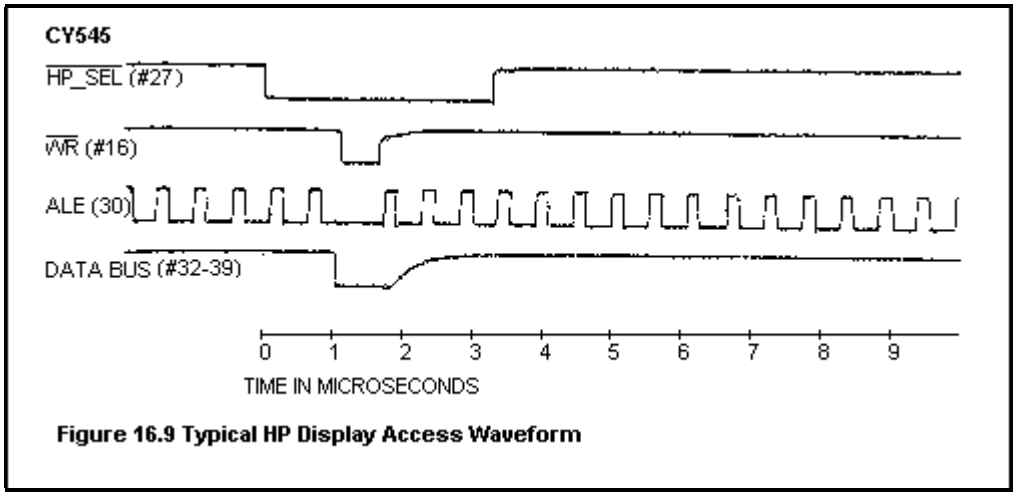


Figure 16.9 Typical HP Display Access Waveform

Thumbwheel Switch Interface Control Signals

The CY545 also supports the reading of parameters from external thumbwheel switches, as discussed previously. The design of the thumbwheel switch interface circuitry is shown in a previous section, and will not be reproduced here. However, we will discuss the CY545 signals used to enable the thumbwheel switches, and show example timing waveforms.

The CY545 also provides all address and strobe timing signals to the switches. Signals required are SW_SEL, ALE, RD, and the data bus lines. The switch address is again multiplexed on the data bus, and requires the external address latch. All 8 bits of the address are used to decode the switch selection logic. Note that the thumbwheel switch interface is a read-only function.

When the CY545 reads the thumbwheel switches, it drives the SW_SEL signal low. Each digit of the parameter is then read by the two stage read operation. The address of the desired digit is first generated, and latched by the ALE signal, then the data bus is floated, and the switch value is enabled during the read strobe signal from RD.

When the entire parameter value has been read, either as 3, 5, or 8 digits, the SW_SEL signal goes high again, disabling the RD strobe for the switches. A typical thumbwheel switch parameter read waveforms follows.

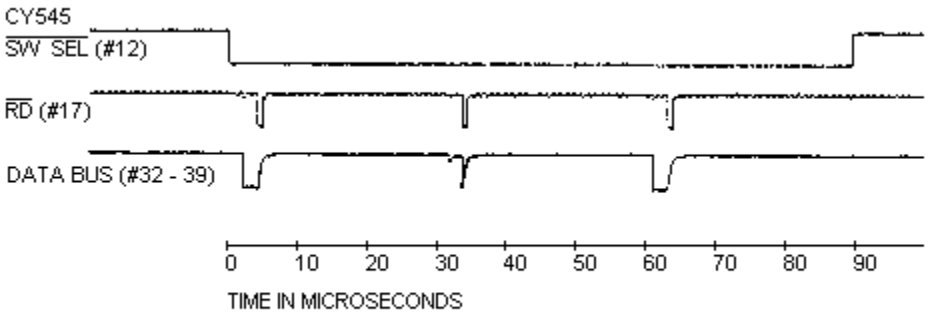


Figure 16.10 Typical Thumbwheel Switch Parameter Read Waveform

Stepper Motor Interface Signals

The CY545 implements eight signals dedicated to the stepper motor, plus the user selectable function bits, which can generate output control signals for special hardware, or be tested for special conditions that guide the CY545 through a program of motions, or be used for home sensor seeking.

PULSE and CWW

The two primary control signals for the motor are PULSE and CCW, which provide the step and direction indications to the motor. These signals are connected to the external power driver that actually runs the motor. Every time the CY545 directs the motor to take a step, a signal is generated on the PULSE line.

The power driver should advance the motor by one increment for each pulse. Note that the step increment could be a full step, half step, quad step, or micro step. This will be determined by the application and the power driver. The CY545 does not know or care what the step size is. All position information is treated in terms of the step increment, not in terms of the linear displacement or degrees of rotation of the motor.

The CCW signal is a level active signal that tells the power driver in which direction to step the motor. When high, the driver should step counter clockwise, and when low, the driver should step clockwise. Note that the physical rotation of the motor will depend on how the motor windings are connected to the power driver, not on the level of the CCW signal. When this signal is high, the CY545 will decrement the current position for every step taken, and when the signal is low, it will increment the current position for every step taken.

Also note that the CCW signal can be used as an input, to force stepping in the desired direction. In this case, the software direction command should be set to “-”, so the CY545 tries to drive the CCW signal high. Now an external circuit can easily override the CY545, and force the CCW signal low, causing stepping in the "+" direction. If this is tried while the CY545 is driving the CCW signal low, a much stronger external circuit will be required to override the CY545, since it has much stronger sink current capability than source current capability. This could ultimately burn out the CCW signal, and should be avoided if possible.

Several PULSE signal waveforms are shown below, for different step rates. Notice that the PULSE signal is normally high, and goes low at the beginning of a step. It stays low for a fixed amount of time, which depends on the crystal frequency of the CY545, and then goes back high for the duration of the step time. When it is time for the next step, PULSE goes low again.

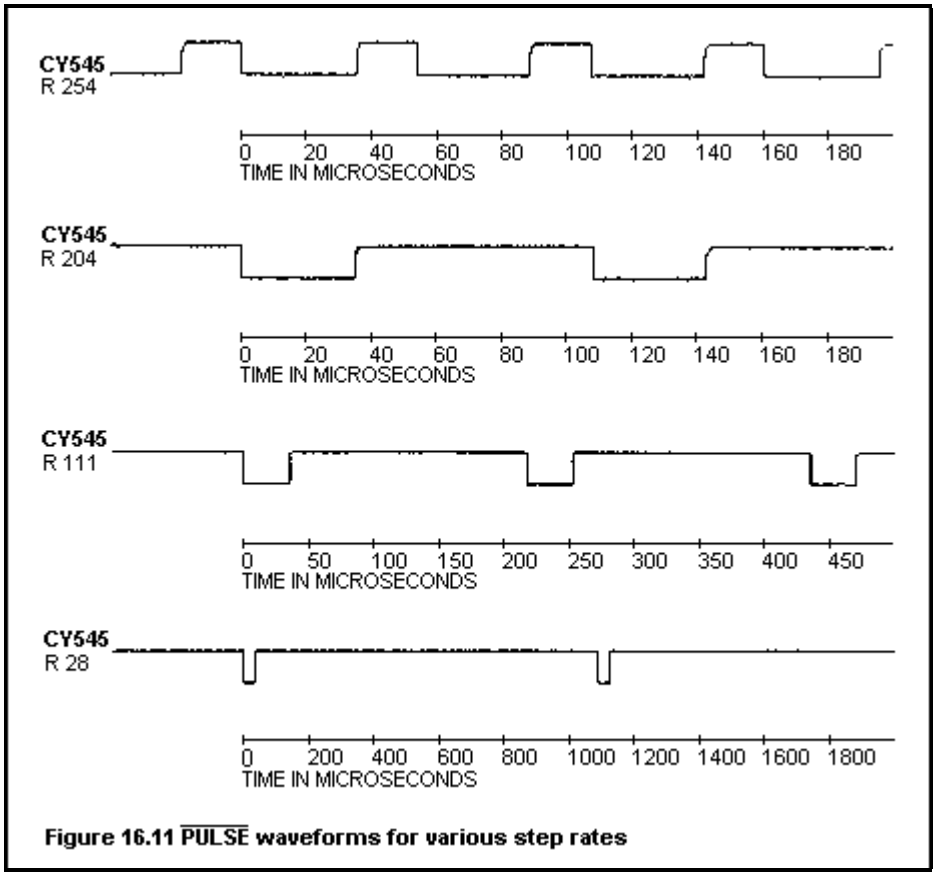


Figure 16.11 $\overline{\text{PULSE}}$ waveforms for various step rates

When the step rate parameters are set to enable acceleration, the CY545 will vary the step rate from the starting rate (First Rate parameter) to the final rate (Rate parameter) for each motion it takes. This means the PULSE waveform will vary from the slower rate to the faster rate as the CY545 accelerates the motor up to the slew rate, and then will be stable for the duration of the slewing period. The CY545 will determine when to begin deceleration so that the motor will slow down again to the starting rate as the desired number of steps has expired.

STOPPED

The STOPPED signal is a status output from the CY545, indicating whether the motor is stepping or not. At the beginning of a motion, from either a Go or Position command, the STOPPED signal is brought low. It will continue to stay low for the duration of that motion. At the end of the motion, the STOPPED signal is brought high again. The low-to-high transition on STOPPED indicates when the motion is over, and the high level indicates that the motor is no longer moving. A typical waveform is shown below.

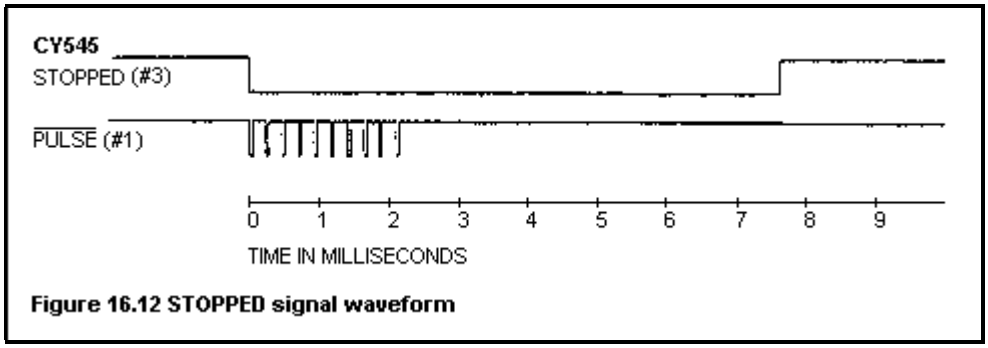


Figure 16.12 STOPPED signal waveform

The STOPPED signal may be used as a motion complete indication to the host system. It may also control the power selection for the stepper motor power driver, switching automatically between a high power selection while stepping, and a parking power selection while stopped. This can provide maximum torque during motions and provide holding torque while the motor is not moving, in order to prevent overheating of the motor. The CY545 automatically delays for 5 milliseconds after the last step, before bringing the STOPPED signal high again. This allows the motor to settle into its final position before the high driving power is removed.

SLEW

Another status output from the CY545 is the SLEW signal, which indicates when the selected step rate (Rate parameter) has been achieved. This signal is normally high, and goes low while the CY545 is stepping the motor without acceleration or deceleration. A typical waveform is shown below. Notice that the time between STOPPED being low and SLEW being low represents the time to accelerate or decelerate the motor.

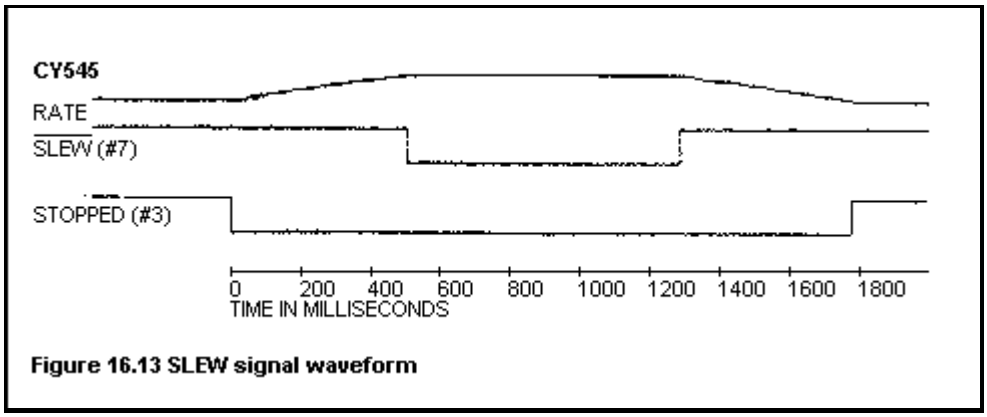


Figure 16.13 SLEW signal waveform

The SLEW signal may also be used as an input. If this signal is driven low before the beginning of a motion, the CY545 will enter the continuous stepping mode. In this mode, it will accelerate

from the starting rate to the slew rate, then run indefinitely at this rate. There are no specific number of steps to move or target position to stop at. However, the current position is still updated during the motion, and may be queried when the CY545 has stopped again. The INHIBIT_ABORT signal is used to end the continuous motion. The SLEW signal is tested just before the CY545 begins the first step of a motion.

Alternatively, the continuous mode may also be selected by the Continuous command. This should be followed by a Go command to actually start the motion. The INHIBIT_ABORT signal is still used to end the motion. Continuous mode stepping is disabled at the end of every motion, so if a second motion is desired in continuous mode, another Continuous command must be issued, or the SLEW signal must be driven low before the motion starts.

INHIBIT_ABORT

The INHIBIT_ABORT signal is a stepping control input to the CY545, with a number of functions. In general, it is used to override the normal stepping behavior of the CY545, as selected by the CY545 commands.

If this signal is low at the beginning of a motion, started by a Position or Go command, that motion will not start until the INHIBIT_ABORT is brought high again. This allows you to synchronize the motion to an external signal, rather than a CY545 command. Multiple axes of motion could be started simultaneously by using this signal to control the start of stepping. An example waveform is shown below.

Note that once a Position or Go command is executed by the CY545, it will wait indefinitely for the INHIBIT_ABORT signal to be high. There is no signal to terminate the motion until it has been allowed to start.

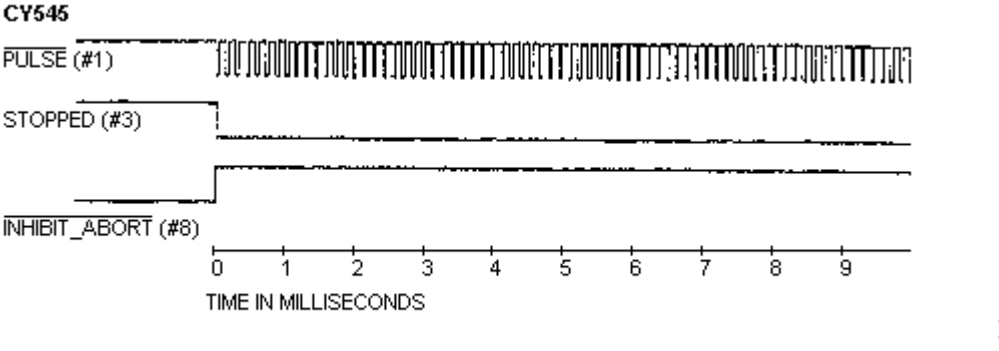


Figure 16.14 INHIBIT_ABORT at the start of motion

Once the motor is moving, the INHIBIT_ABORT signal may be used to slow down and stop a motion before the CY545 reaches the target position. Using the signal in this way provides a controlled early stop to the motion.

If INHIBIT_ABORT is driven low during a motion, the CY545 switches to a deceleration mode. It will decelerate symmetrically to the acceleration, until it achieves the selected initial stepping rate (First Rate parameter). The CY545 will then test the INHIBIT_ABORT signal once for each step, and if it is low, the motion will stop. Otherwise, the motion will continue, at the starting rate, until the target position is reached. To slow down the motor and stop when the initial step rate is reached, simply hold the INHIBIT_ABORT signal low until the CY545 brings the STOPPED signal high again. Several cases are illustrated below.

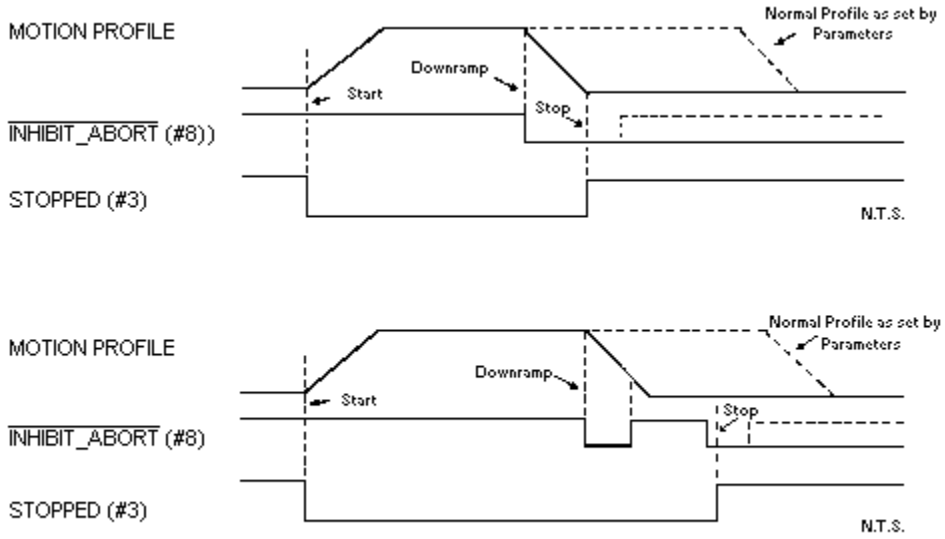


Figure 16.15 Stopping motions with INHIBIT_ABORT

CW_LIMIT and CCW_LIMIT

The CY545 also provides two inputs for immediately stopping a motion, the CW_LIMIT and CCW_LIMIT signals. If the relevant signal goes low during a motion in that direction, the CY545 immediately stops the motion. There will be no deceleration before the stop; the CY545 simply abandons the motion.

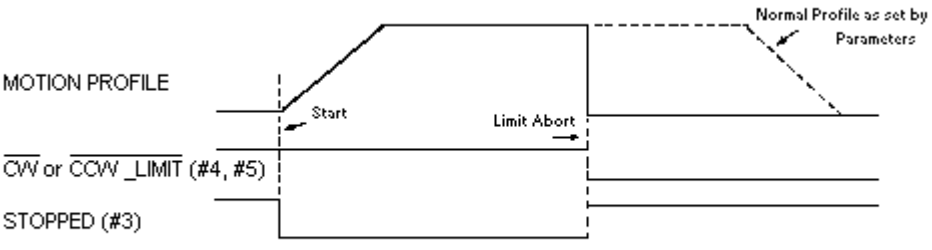


Figure 16.16 CW_LIMIT or CCW_LIMIT signals stop a motion

Also, if a limit signal is low at the beginning of a motion, the motion will not be taken. The CY545 immediately aborts the motion without taking any steps. Normal motions are allowed in the opposite direction. Example waveforms are shown above.

***** Warning *** Warning *** Warning *****

Although the CW_LIMIT and the CCW_LIMIT signals allow the CY545 to provide some protection to systems that accidentally run into limits, they should NOT be used as the only protection if reaching the limits could endanger operators or damage the equipment.

Unexpectedly hitting a limit implies some failure, and no component, including the host computer or CY545, can be fully trusted in this situation. Automatic recovery from a limit should be used with extreme caution, since the system could perform the recovery, then run back into the limit again, if the original cause of the failure has not been fixed.

The best protection in cases where the system has failed, is to disable the power to the stepper motor drivers. This will allow the motors to stop moving the load, even if they are commanded to continue stepping. The cause of failure should then be determined and fixed before the motors are energized again.

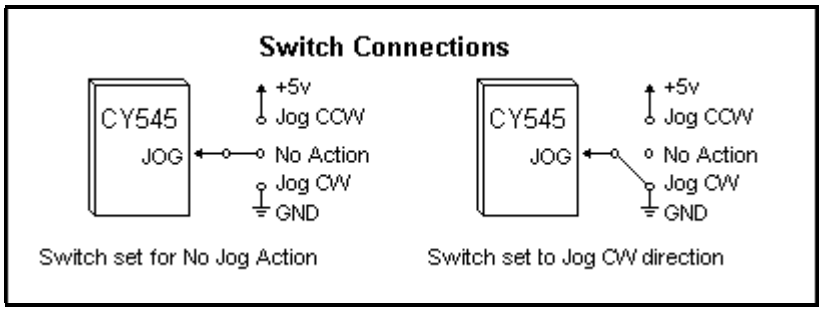
JOG Mode Operation

The CY545 provides manual stepper control through the JOG signal. This signal is tested when the CY545 is not executing a command or performing another motion. This is a special signal, because it is checked for one of three different states and is both an input and an output.

When the line is left floating, so the CY545 can drive it both ways, the jog function is disabled, and stepping only occurs when the CY545 is commanded to take a motion.

When the line is driven low, the CY545 will jog the motor in the clockwise direction. The CCW signal will change state, if required, to select this direction.

When the line is driven high, the CY545 will jog the motor in the counter clockwise direction. Again, the CCW signal will change state, if required, to select this direction. Note that the signal level on the JOG line matches the stepping direction selected by the CCW signal.



The CCW signal is restored to its original state when each jog single-step finishes.

Normally, the JOG signal will be connected to a three position, center-off switch. One side of the switch will be connected to ground, while the other side is connected to +5 volts. Jogging is enabled in each direction by pushing the switch from the center position to one side or the other. This is shown in the figure.

When jogging, the CY545 will take one step, then check for any new command inputs. If a new command is available, it will be executed, otherwise the JOG signal will be tested again. When the CY545 is idle, the effect of driving the JOG line is for the CY545 to take repeated single steps. Note that the checking for new commands during jogging allows you to issue CY545

commands between jog steps, providing jog motion and command execution at the same time! However, the Jog signal is ignored while these commands are executing.

While jogging, the CY545 will use a step rate derived from the parameter of the First Rate command. The normal initial stepping rate for continuous motions is taken from the First Rate parameter value, and this rate is divided by 20. Note that the same step rate is used by the Home command. The CY545 Step Rate Table lists the step rates for continuous motion. These rates should be divided by 20 to obtain the jogging step rate. With a 12 MHz clock, jog rates from about 1 step per second to about 1000 steps per second are possible.

The CY545 will not accelerate or decelerate during jogging, running only at the rate derived from the current value of the First Rate parameter. However, the interactive command execution during jogging does allow you to modify the step rate, by issuing a First Rate command with a new parameter value.

Automatic Position Display

Also, when bit 4 of the Mode Register is set, the CY545 will display the current position at the end of every jog or Home command step. The position is output to the currently enabled display device, and will affect the step rate. For example, sending the current position to a serial device would require enough time to issue the 11 character position string at the baud rate defined for the serial interface.

The JOG signal also represents a minimal mechanism for testing the CY545 and motor subsystem. Without any commands, it is possible to move the motor in either direction, verifying that the PULSE and CCW signals are correctly connected to the motor power driver, and that the motor is properly connected. This also verifies that the CY545 is functioning at least in a very basic fashion.

Home Signal

The CY545 supports an additional stepping function, but it uses the user selectable function signals to implement the process. This function is the automatic home sensor seek, and is selected by the Home command.

A single argument to the Home command specifies which user selectable bit or bits to monitor during the step to home. The parameter value is treated the same as for the Til and Wait commands, and is reproduced below:

00H to 07H	Test one of user bits 0 to 7 for 1
08H to 0FH	Test one of data bus bits 0 to 7 for 1
10H to 17H	Test one of user bits 0 to 7 for 0
18H to 1FH	Test one of data bus bits 0 to 7 for 0
80H to 0BFH	Test user bits 0 to 5 for a group match

You have the option of checking any one signal for the home sensor, or using the six-signal group of user bits 0 to 5. Normally, a single home sensor would be connected to a single user bit signal.

This command uses the same step rate as the jog function described above, a division by 20 of the step rate selected by the First Rate parameter value, so the CY545 will step the motor at about 1 to 1000 steps per second while seeking the home sensor signal.

The Home command starts by testing the specified bit pattern for a match with the external signals. If there is a match, the CY545 will begin by stepping in the CCW direction, and will continue in this direction until there is no longer a match. The CCW stepping phase is skipped if the initial test indicates no match between the argument bit pattern and the external signals.

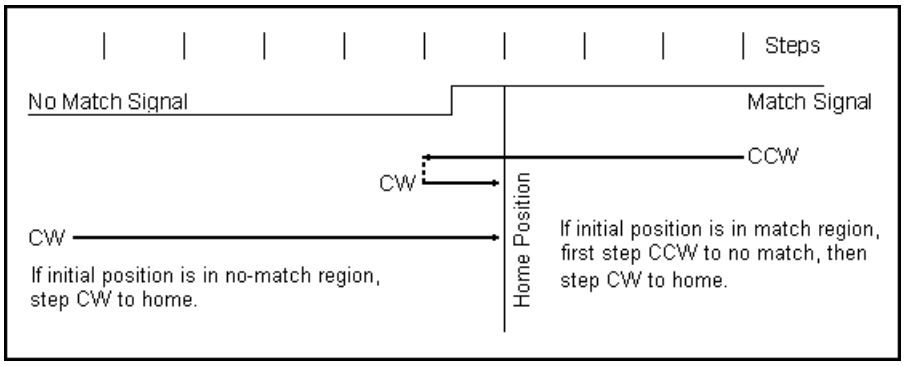
When there is no match between the bit parameter and the external signals, the CY545 steps in the CW direction. It will continue to step in this direction until the signals match the parameter value. At this point the CY545 stops, and sets the current position to zero.

Thus, the Home command steps the motor to seek the edge between a match and no match condition on the home signal. The CY545 then stops at the first step where a match occurs, and is always stepping in the CW direction looking for this match. This mechanism should always seek the same mechanical position, since any directional backlash is compensated for by using the same final step direction in seeking home.

If the CY545 runs into one of the limits while looking for the home signal, it will abort the home command and stop stepping. The current position is not changed to zero in this case. As with the jog function, the current position will be displayed after every step, if bit 4 of the Mode Register is set.

Some care must be used in designing the home sensor signal for the CY545, to insure that it initially steps in the correct direction to seek the home signal. If the application involves a circular positioning function, in which the mechanism rotates a wheel or turntable, there is no special consideration, since the mechanical positions will always repeat after some number of steps. The CY545 can find the home sensor signal no matter what direction it starts to step.

However, if the application involves a linear displacement, the mechanical home position can only be found by stepping in one direction, depending on the relative position between the mechanical home and the current position. In this case, the sensor signal should be designed to indicate the direction to home, as well as the actual home position, found by the change in signal from no match to match conditions. This is illustrated below:



If the home sensor signal cannot be designed to indicate the initial direction to home, the motor must be positioned so that it is on the “correct” side of home before the Home command is used. This will insure that the CY545 always steps towards the sensor on seeking the home position. The CW and CCW limits might be useful in this situation, since you could move to a known limit, then use the home command to step in the opposite direction until the home position is found.

CHAPTER 17 - CY545 STEP RATE INFORMATION

CY545 Step Rate Information

Step rates for the CY545 are chosen from the CY545 Step Rate Table, which appears at the end of this section. Selections for the starting rate are made by the value of the First Rate (F) parameter, and selections for the slewing rate are made by the value of the Rate (R) parameter.

Each rate parameter value acts as an index into the CY545 Step Rate Table, which has 256 unique entries, ranging from a parameter value of zero to 255. Thus, for any crystal clock frequency, the CY545 can generate 256 different step rates. The CY545 Step Rate Table is shown for a clock frequency of 12 MHz, so the specified rates must be linearly scaled by the actual clock frequency when a different clock is used. The scaling factor is Fcy/12, where Fcy is the operating frequency, in megahertz.

As an example, the following commands would select a starting rate of 200 steps per second, and a final rate of 10000 steps per second, if the CY545 is operated at 12 MHz.

F 8 200 steps per second initial rate
R 203 10000 steps per second final rate

The Slope Parameter

When the CY545 is commanded to step through a motion, by either the Go or Position commands, stepping will start at the rate selected by the F parameter. The rate will then increase regularly through the other rates in the table, until the final rate, selected by the R parameter is achieved. The CY545 will then slew at this rate until it is time to decelerate back to the selected starting rate as the specified number of steps is taken.

If the travel distance is too short to achieve the final rate, the CY545 will perform a partial acceleration, going to the highest rate possible before decelerating back to the starting rate.

The speed at which the CY545 changes from one rate to the next is selected by the Slope (S) parameter. The value of S can range from 1 to 255, with 1 representing the slowest acceleration, and 255 representing the fastest acceleration.

For any particular application, the stepping parameter values, including the slope, are best determined experimentally when you are looking for the operating limits of the system. Each motor size, driver type, and load combination will have different performance limits. In general, the required range of step rates is known for a particular application, and will determine the values for F and R, but the acceleration value is not so easily specified.

Optimal Acceleration Curves

The stepping algorithm used in the CY545 is a derivative of our CY525 algorithm, which has linear to optimal acceleration behavior over much of its range. In fact, the acceleration curve is near optimal in shape up to about 10000 steps per second. This will provide maximum performance for any motor/driver/load combination. An example waveform is shown below:

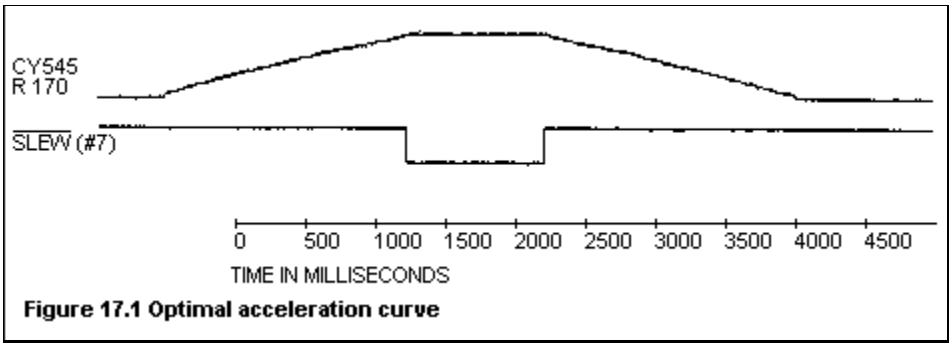


Figure 17.1 Optimal acceleration curve

Acceleration Curve as a Function of R

Another feature of the CY545 acceleration algorithm is that the shape is fixed by the value of the Slope parameter (S). The acceleration curve is always the same for a given value of S. The step rate parameters merely determine where the CY545 starts stepping, and where it stops accelerating and starts slewing in the motion profile. The three parameters F, R, and S, are not coupled by the stepping algorithm, so each may be specified independently. These features are illustrated in the figure below.

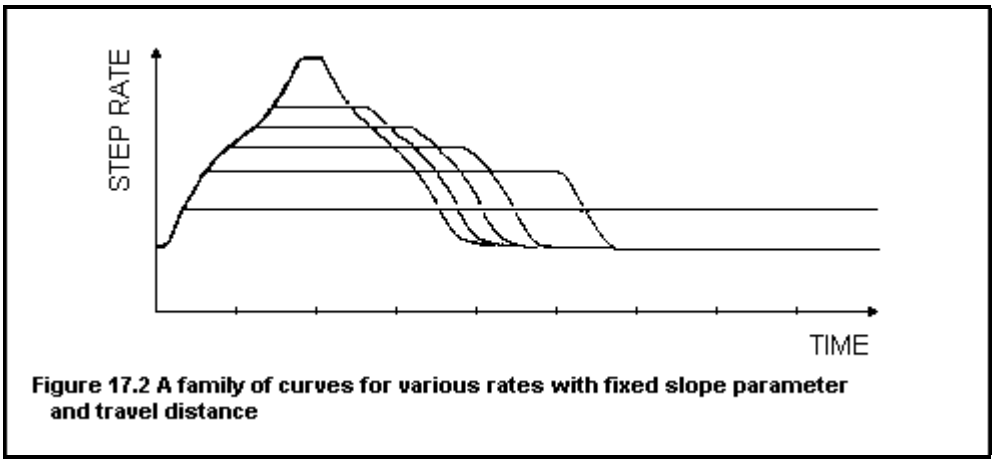


Figure 17.2 A family of curves for various rates with fixed slope parameter and travel distance

Slope and Elapsed Time

While it is difficult to provide an exact formula for the CY545 acceleration behavior, due to the complex nature of the stepping algorithm, we can provide an approximate formula. This should be used as a general guide only, with actual values determined experimentally. The formula is given by:

$$\text{accel} = a(s) = \frac{dV(s)}{dT(s)} = \frac{(3906 * D)}{(256 - s)} \quad @12\ \text{MHz}$$

where

a(s) is the acceleration in steps/second^2
s is the slope parameter value
D is the "distance" between two adjacent rates in steps/second, the incremental slope

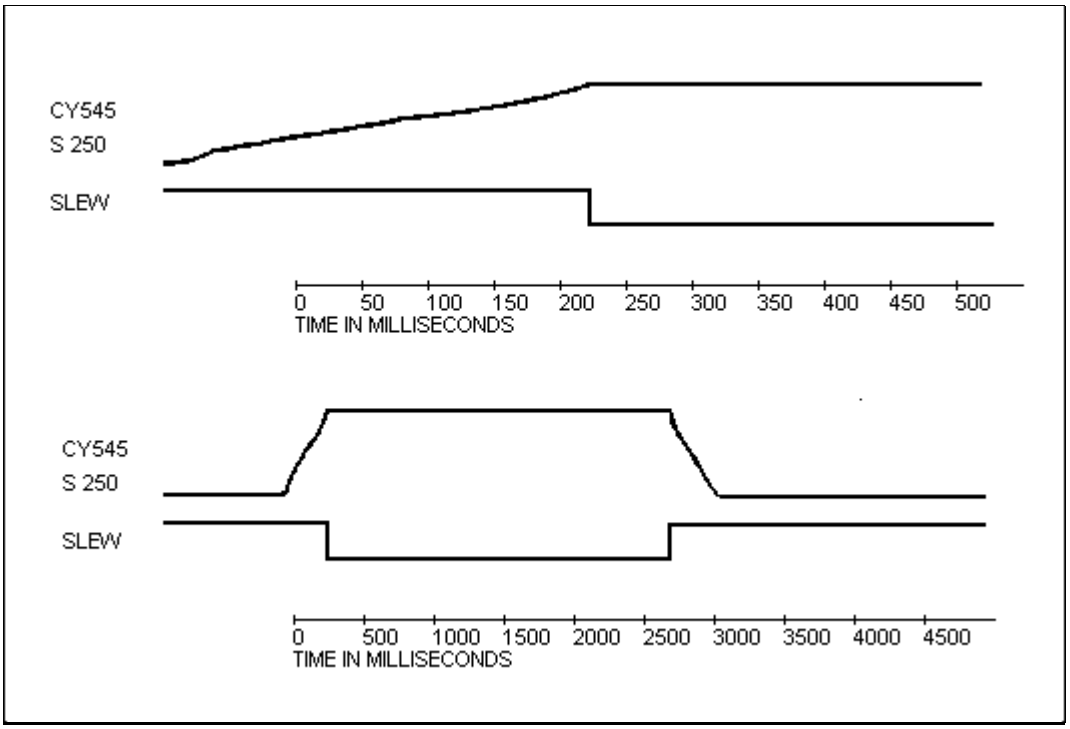
also, the ramp time is given by the acceleration and the dynamic range of step rates taken as:

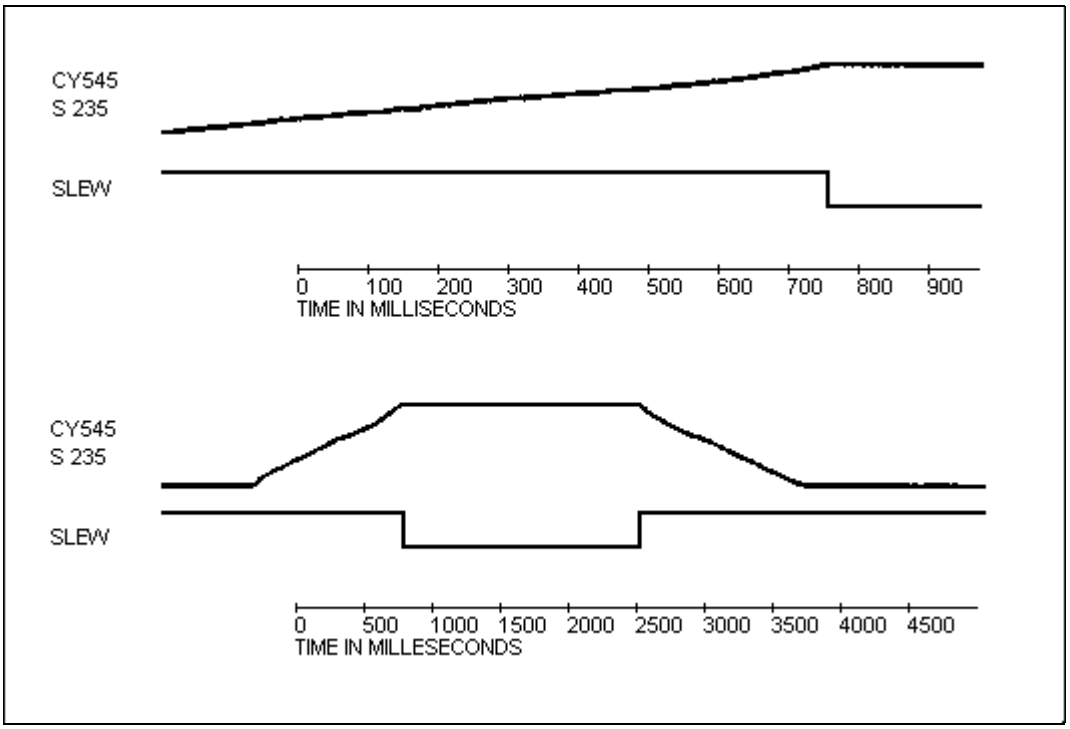
$$t = \frac{|Rmax - R0|}{a(s)} = \frac{|Rmax - R0|}{\frac{3906 * D)}{(256 - s)}}$$

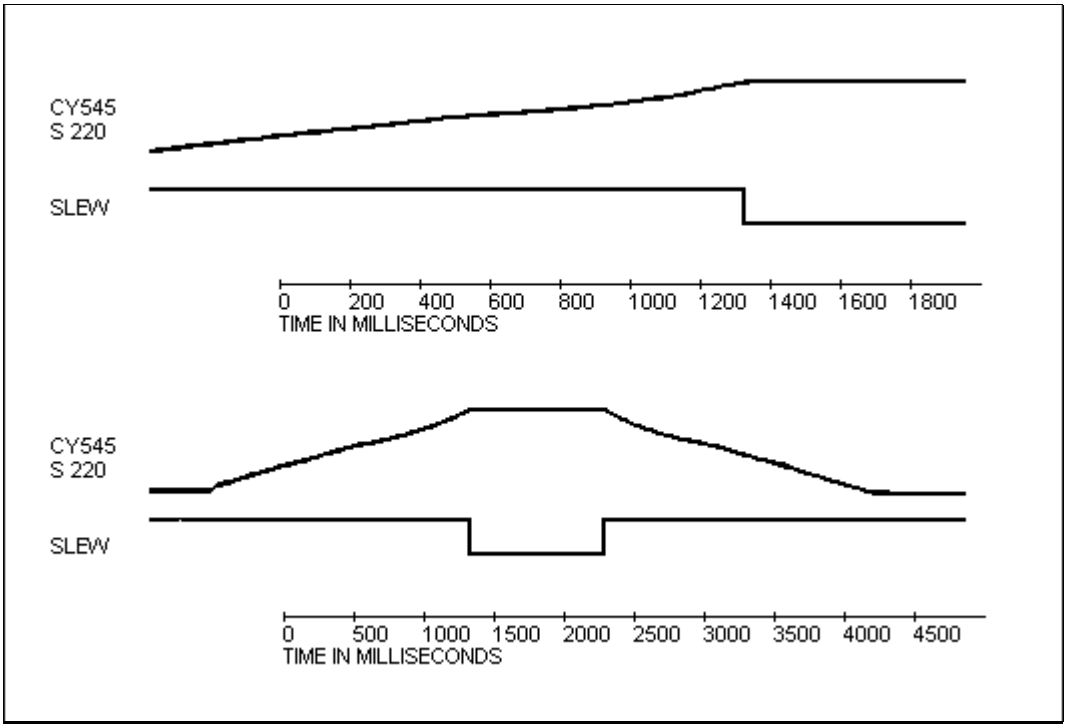
The value used for D in the above formulas must be chosen with care, since the distance between rate points changes as you move through the CY545 Step Rate Table. D starts with a value of 25, then 50, then 35, and finally inflects to larger values as we reach the digital resolution limit of the CY545.

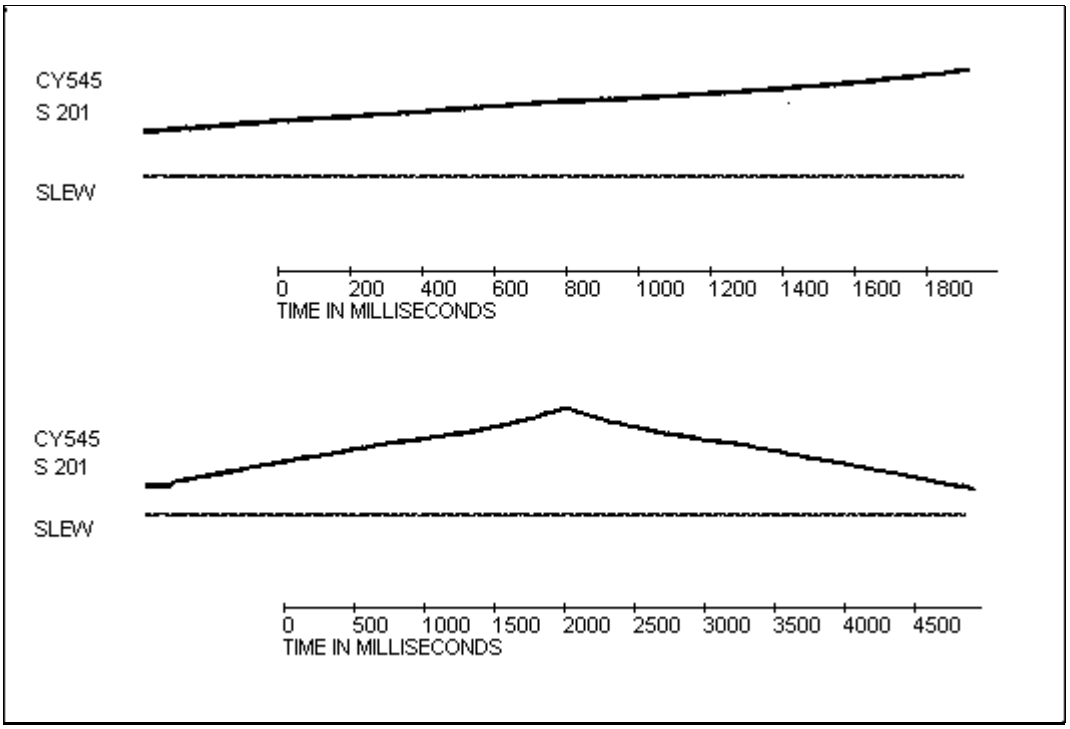
The following figures show typical performance under various values of slope, and may be used as a general guide in selecting initial operating parameters. The figures are followed by the CY545 Step Rate Table, which lists all 256 possible step rates when operating the CY545 with a 12 MHz clock.

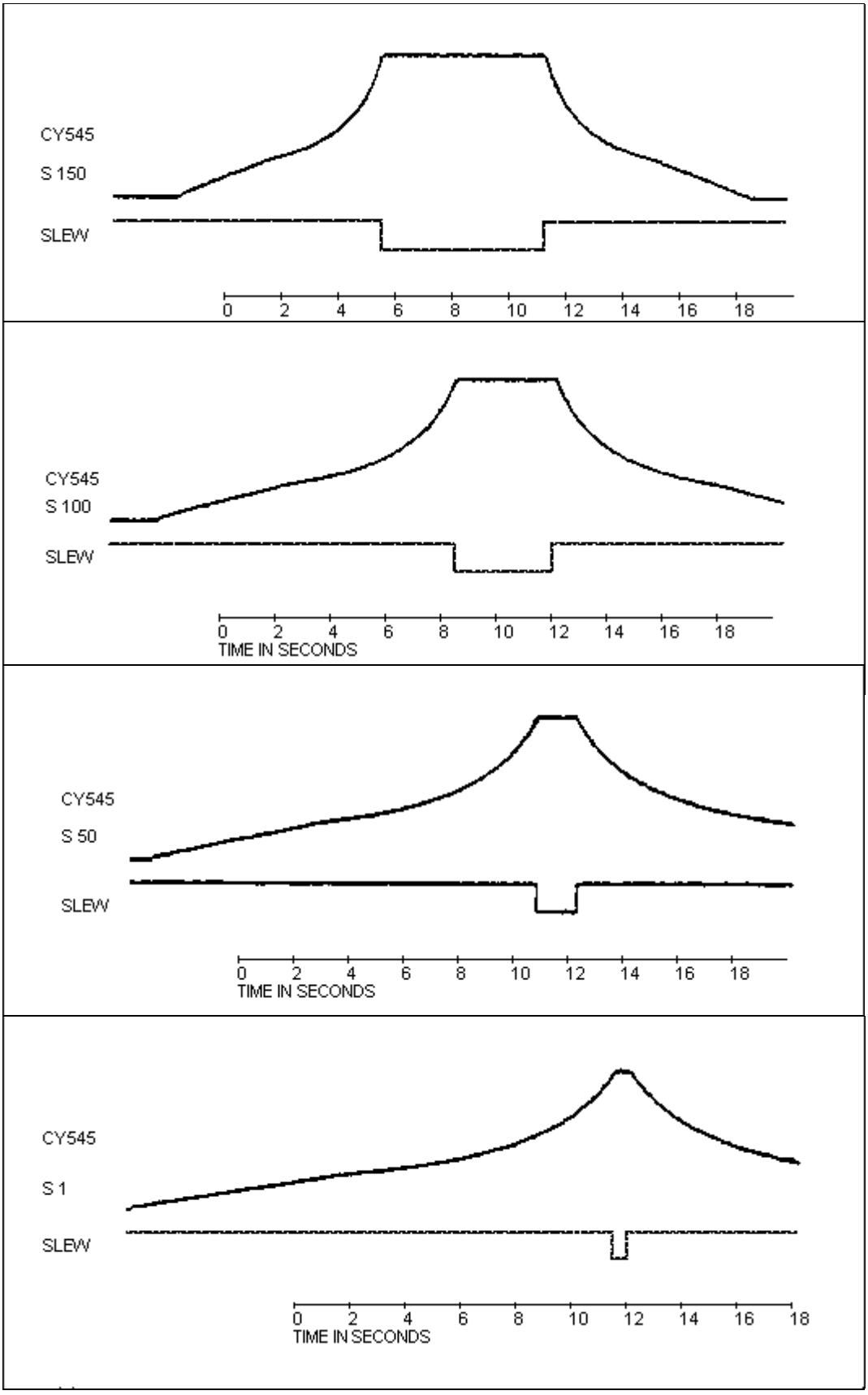
CY545 Slope Curves






CY545
S 220
SLEW
0 200 400 600 800 1000 1200 1400 1600 1800
TIME IN MILLISECONDS
CY545
S 220
SLEW
0 500 1000 1500 2000 2500 3000 3500 4000 4500
TIME IN MILLISECONDS


CY545
S 201
SLEW
0 200 400 600 800 1000 1200 1400 1600 1800
TIME IN MILLISECONDS
CY545
S 201
SLEW
0 500 1000 1500 2000 2500 3000 3500 4000 4500
TIME IN MILLISECONDS


CY545
S 150
SLEW
0 2 4 6 8 10 12 14 16 18
CY545
S 100
SLEW
0 2 4 6 8 10 12 14 16 18
TIME IN SECONDS
CY545
S 50
SLEW
0 2 4 6 8 10 12 14 16 18
TIME IN SECONDS
CY545
S 1
SLEW
0 2 4 6 8 10 12 14 16 18
TIME IN SECONDS

CY545 Step Rate Table @ 11 MHz

R 0	18	R 43	1605	R 86	3485	R 128	5238	R 171	6944	R 214	10300
R 1	22	R 44	1649	R 87	3526	R 129	5268	R 172	6997	R 215	10417
R 2	46	R 45	1694	R 88	3567	R 130	5299	R 173	7051	R 216	10536
R 3	69	R 46	1743	R 89	3609	R 131	5329	R 174	7106	R 217	10659
R 4	92	R 47	1787	R 90	3652	R 132	5361	R 175	7161	R 218	10784
R 5	115	R 48	1833	R 91	3696	R 133	5392	R 176	7218	R 219	10913
R 6	137	R 49	1878	R 92	3741	R 134	5424	R 177	7275	R 220	11044
R 7	160	R 50	1926	R 93	3788	R 135	5456	R 178	7333	R 221	11179
R 8	183	R 51	1971	R 94	3835	R 136	5489	R 179	7392	R 222	11317
R 9	206	R 52	2015	R 95	3884	R 137	5522	R 180	7453	R 223	11458
R 10	229	R 53	2065	R 96	3934	R 138	5556	R 181	7514	R 224	11603
R 11	252	R 54	2107	R 97	3986	R 139	5589	R 182	7576	R 225	11752
R 12	275	R 55	2152	R 98	4038	R 140	5624	R 183	7639	R 226	11905
R 13	298	R 56	2198	R 99	4092	R 141	5658	R 184	7703	R 227	12061
R 14	321	R 57	2247	R 100	4148	R 142	5694	R 185	7768	R 228	12222
R 15	344	R 58	2292	R 101	4186	R 143	5729	R 186	7835	R 229	12387
R 16	367	R 59	2338	R 102	4224	R 144	5765	R 187	7902	R 230	12557
R 17	412	R 60	2381	R 103	4264	R 145	5802	R 188	7971	R 231	12731
R 18	459	R 61	2431	R 104	4304	R 146	5839	R 189	8041	R 232	12911
R 19	504	R 62	2477	R 105	4344	R 147	5876	R 190	8112	R 233	13095
R 20	550	R 63	2518	R 106	4386	R 148	5914	R 191	8185	R 234	13285
R 21	596	R 64	2568	R 107	4428	R 149	5952	R 192	8258	R 235	13480
R 22	641	R 65	2612	R 108	4472	R 150	5991	R 193	8333	R 236	13682
R 23	688	R 66	2657	R 109	4516	R 151	6031	R 194	8410	R 237	13889
R 24	733	R 67	2704	R 110	4561	R 152	6071	R 195	8488	R 238	14103
R 25	780	R 68	2753	R 111	4606	R 153	6111	R 196	8567	R 239	14323
R 26	825	R 69	2795	R 112	4653	R 154	6152	R 197	8648	R 240	14550
R 27	871	R 70	2838	R 113	4701	R 155	6194	R 198	8730	R 241	14785
R 28	917	R 71	2892	R 114	4750	R 156	6236	R 199	8814	R 242	15027
R 29	963	R 72	2938	R 115	4799	R 157	6279	R 200	8900	R 243	15278
R 30	1008	R 73	2976	R 116	4850	R 158	6322	R 201	8987	R 244	15537
R 31	1054	R 74	3025	R 117	4902	R 159	6366	R 202	9076	R 245	15805
R 32	1100	R 75	3066	R 118	4955	R 160	6410	R 203	9167	R 246	16082
R 33	1146	R 76	3118	R 119	4982	R 161	6455	R 204	9259	R 247	16369
R 34	1192	R 77	3161	R 120	5009	R 162	6501	R 205	9354	R 248	16667
R 35	1237	R 78	3194	R 121	5037	R 163	6548	R 206	9450	R 249	16975
R 36	1284	R 79	3228	R 122	5064	R 164	6595	R 207	9549	R 250	17296
R 37	1329	R 80	3262	R 123	5093	R 165	6643	R 208	9649	R 251	17628
R 38	1374	R 81	3297	R 124	5121	R 166	6691	R 209	9752	R 252	17974
R 39	1421	R 82	3333	R 125	5150	R 167	6740	R 210	9857	R 253	18333
R 40	1467	R 83	3370	R 126	5179	R 168	6790	R 211	9964	R 254	18707
R 41	1513	R 84	3408	R 127	5208	R 169	6841	R 212	10073	R 255	19097
R 42	1559	R 85	3446			R 170	6892	R 213	10185		

CY545 Step Rate Table @ 12 MHz

R 0	20	R 43	1751	R 86	3802	R 128	5714	R 171	7576	R 214	11236
R 1	25	R 44	1799	R 87	3846	R 129	5747	R 172	7634	R 215	11364
R 2	50	R 45	1848	R 88	3891	R 130	5780	R 173	7692	R 216	11494
R 3	75	R 46	1901	R 89	3937	R 131	5814	R 174	7752	R 217	11628
R 4	100	R 47	1949	R 90	3984	R 132	5848	R 175	7812	R 218	11765
R 5	125	R 48	2000	R 91	4032	R 133	5882	R 176	7874	R 219	11905
R 6	150	R 49	2049	R 92	4082	R 134	5917	R 177	7937	R 220	12048
R 7	175	R 50	2101	R 93	4132	R 135	5952	R 178	8000	R 221	12195
R 8	200	R 51	2151	R 94	4184	R 136	5988	R 179	8065	R 222	12346
R 9	225	R 52	2198	R 95	4237	R 137	6024	R 180	8130	R 223	12500
R 10	250	R 53	2252	R 96	4292	R 138	6061	R 181	8197	R 224	12658
R 11	275	R 54	2299	R 97	4348	R 139	6098	R 182	8264	R 225	12821
R 12	300	R 55	2347	R 98	4405	R 140	6135	R 183	8333	R 226	12987
R 13	325	R 56	2398	R 99	4464	R 141	6173	R 184	8403	R 227	13158
R 14	350	R 57	2451	R 100	4525	R 142	6211	R 185	8475	R 228	13333
R 15	375	R 58	2500	R 101	4566	R 143	6250	R 186	8547	R 229	13514
R 16	400	R 59	2551	R 102	4608	R 144	6289	R 187	8621	R 230	13699
R 17	450	R 60	2597	R 103	4651	R 145	6329	R 188	8696	R 231	13889
R 18	500	R 61	2653	R 104	4695	R 146	6369	R 189	8772	R 232	14085
R 19	550	R 62	2703	R 105	4739	R 147	6410	R 190	8850	R 233	14286
R 20	600	R 63	2747	R 106	4785	R 148	6452	R 191	8929	R 234	14493
R 21	650	R 64	2801	R 107	4831	R 149	6494	R 192	9009	R 235	14706
R 22	700	R 65	2849	R 108	4878	R 150	6536	R 193	9091	R 236	14925
R 23	750	R 66	2899	R 109	4926	R 151	6579	R 194	9174	R 237	15152
R 24	799	R 67	2950	R 110	4975	R 152	6623	R 195	9259	R 238	15385
R 25	851	R 68	3003	R 111	5025	R 153	6667	R 196	9346	R 239	15625
R 26	900	R 69	3049	R 112	5076	R 154	6711	R 197	9434	R 240	15873
R 27	950	R 70	3096	R 113	5128	R 155	6757	R 198	9524	R 241	16129
R 28	1000	R 71	3155	R 114	5181	R 156	6803	R 199	9615	R 242	16393
R 29	1050	R 72	3205	R 115	5236	R 157	6849	R 200	9709	R 243	16667
R 30	1100	R 73	3247	R 116	5291	R 158	6897	R 201	9804	R 244	16949
R 31	1149	R 74	3300	R 117	5348	R 159	6944	R 202	9901	R 245	17241
R 32	1200	R 75	3344	R 118	5405	R 160	6993	R 203	10000	R 246	17544
R 33	1250	R 76	3401	R 119	5435	R 161	7042	R 204	10101	R 247	17857
R 34	1300	R 77	3448	R 120	5464	R 162	7092	R 205	10204	R 248	18182
R 35	1350	R 78	3484	R 121	5495	R 163	7143	R 206	10309	R 249	18519
R 36	1401	R 79	3521	R 122	5525	R 164	7194	R 207	10417	R 250	18868
R 37	1449	R 80	3559	R 123	5556	R 165	7246	R 208	10526	R 251	19231
R 38	1499	R 81	3597	R 124	5587	R 166	7299	R 209	10638	R 252	19608
R 39	1550	R 82	3636	R 125	5618	R 167	7353	R 210	10753	R 253	20000
R 40	1600	R 83	3676	R 126	5650	R 168	7407	R 211	10870	R 254	20408
R 41	1650	R 84	3717	R 127	5682	R 169	7463	R 212	10989	R 255	20833
R 42	1701	R 85	3759			R 170	7519	R 213	11111		

CY545 Step Rate Table @ 14.75 MHz

R 0	24	R 43	2153	R 86	4674	R 128	7024	R 171	9312	R 214	13811
R 1	30	R 44	2211	R 87	4728	R 129	7064	R 172	9383	R 215	13968
R 2	61	R 45	2272	R 88	4783	R 130	7105	R 173	9455	R 216	14128
R 3	92	R 46	2337	R 89	4839	R 131	7146	R 174	9528	R 217	14293
R 4	123	R 47	2396	R 90	4897	R 132	7188	R 175	9603	R 218	14461
R 5	154	R 48	2458	R 91	4956	R 133	7230	R 176	9678	R 219	14633
R 6	184	R 49	2519	R 92	5017	R 134	7273	R 177	9755	R 220	14809
R 7	215	R 50	2582	R 93	5079	R 135	7316	R 178	9833	R 221	14990
R 8	246	R 51	2643	R 94	5143	R 136	7360	R 179	9913	R 222	15175
R 9	277	R 52	2701	R 95	5208	R 137	7405	R 180	9993	R 223	15365
R 10	307	R 53	2768	R 96	5275	R 138	7449	R 181	10075	R 224	15559
R 11	338	R 54	2826	R 97	5344	R 139	7495	R 182	10158	R 225	15759
R 12	369	R 55	2885	R 98	5415	R 140	7541	R 183	10243	R 226	15963
R 13	399	R 56	2948	R 99	5487	R 141	7587	R 184	10329	R 227	16173
R 14	430	R 57	3013	R 100	5562	R 142	7635	R 185	10417	R 228	16389
R 15	461	R 58	3073	R 101	5613	R 143	7682	R 186	10506	R 229	16610
R 16	492	R 59	3136	R 102	5664	R 144	7731	R 187	10596	R 230	16838
R 17	553	R 60	3193	R 103	5717	R 145	7780	R 188	10688	R 231	17072
R 18	615	R 61	3260	R 104	5771	R 146	7829	R 189	10782	R 232	17312
R 19	676	R 62	3322	R 105	5825	R 147	7879	R 190	10878	R 233	17560
R 20	737	R 63	3377	R 106	5881	R 148	7930	R 191	10975	R 234	17814
R 21	799	R 64	3443	R 107	5938	R 149	7982	R 192	11074	R 235	18076
R 22	860	R 65	3502	R 108	5996	R 150	8034	R 193	11174	R 236	18346
R 23	922	R 66	3563	R 109	6055	R 151	8087	R 194	11277	R 237	18624
R 24	983	R 67	3626	R 110	6115	R 152	8140	R 195	11381	R 238	18910
R 25	1046	R 68	3691	R 111	6177	R 153	8194	R 196	11488	R 239	19206
R 26	1106	R 69	3747	R 112	6239	R 154	8249	R 197	11596	R 240	19511
R 27	1167	R 70	3805	R 113	6303	R 155	8305	R 198	11706	R 241	19825
R 28	1229	R 71	3877	R 114	6369	R 156	8362	R 199	11819	R 242	20150
R 29	1291	R 72	3940	R 115	6435	R 157	8419	R 200	11934	R 243	20486
R 30	1352	R 73	3991	R 116	6504	R 158	8477	R 201	12051	R 244	20833
R 31	1413	R 74	4057	R 117	6573	R 159	8536	R 202	12170	R 245	21193
R 32	1476	R 75	4111	R 118	6644	R 160	8596	R 203	12292	R 246	21564
R 33	1536	R 76	4181	R 119	6680	R 161	8656	R 204	12416	R 247	21949
R 34	1598	R 77	4239	R 120	6717	R 162	8717	R 205	12543	R 248	22348
R 35	1659	R 78	4283	R 121	6754	R 163	8780	R 206	12672	R 249	22762
R 36	1722	R 79	4328	R 122	6791	R 164	8843	R 207	12804	R 250	23192
R 37	1781	R 80	4374	R 123	6829	R 165	8907	R 208	12939	R 251	23638
R 38	1843	R 81	4421	R 124	6867	R 166	8972	R 209	13076	R 252	24101
R 39	1906	R 82	4470	R 125	6905	R 167	9038	R 210	13217	R 253	24583
R 40	1967	R 83	4519	R 126	6944	R 168	9105	R 211	13361	R 254	25085
R 41	2028	R 84	4569	R 127	6984	R 169	9173	R 212	13507	R 255	25608
R 42	2090	R 85	4621			R 170	9242	R 213	13657		

CY545 Step Rate Table @ 16 MHz

R 0	26	R 43	2335	R 86	5070	R 128	7619	R 170	10025	R 213	14815
R 1	33	R 44	2398	R 87	5128	R 129	7663	R 171	10101	R 214	14981
R 2	67	R 45	2465	R 88	5188	R 130	7707	R 172	10178	R 215	15152
R 3	100	R 46	2535	R 89	5249	R 131	7752	R 173	10256	R 216	15326
R 4	133	R 47	2599	R 90	5312	R 132	7797	R 174	10336	R 217	15504
R 5	167	R 48	2667	R 91	4376	R 133	7843	R 175	10417	R 218	15686
R 6	200	R 49	2732	R 92	5442	R 134	7890	R 176	10499	R 219	15873
R 7	233	R 50	2801	R 93	5510	R 135	7937	R 177	10582	R 220	16064
R 8	267	R 51	2867	R 94	5579	R 136	7984	R 178	10667	R 221	16260
R 9	300	R 52	2930	R 95	5650	R 137	8032	R 179	10753	R 222	16461
R 10	333	R 53	3003	R 96	5722	R 138	8081	R 180	10840	R 223	16667
R 11	367	R 54	3065	R 97	5797	R 139	8130	R 181	10929	R 224	16878
R 12	400	R 55	3130	R 98	5874	R 140	8180	R 182	11019	R 225	17094
R 13	433	R 56	3197	R 99	5952	R 141	8230	R 183	11111	R 226	17316
R 14	467	R 57	3268	R 100	6033	R 142	8282	R 184	11204	R 227	17544
R 15	500	R 58	3333	R 101	6088	R 143	8333	R 185	11299	R 228	17778
R 16	534	R 59	3401	R 102	6144	R 144	8386	R 186	11396	R 229	18018
R 17	600	R 60	3463	R 103	6202	R 145	8439	R 187	11494	R 230	18265
R 18	667	R 61	3537	R 104	6260	R 146	8493	R 188	11594	R 231	18519
R 19	733	R 62	3604	R 105	6319	R 147	8547	R 189	11696	R 232	18779
R 20	800	R 63	3663	R 106	6380	R 148	8602	R 190	11799	R 233	19048
R 21	866	R 64	3735	R 107	6441	R 149	8658	R 191	11905	R 234	19324
R 22	933	R 65	3799	R 108	6504	R 150	8715	R 192	12012	R 235	19608
R 23	1000	R 66	3865	R 109	6568	R 151	8772	R 193	12121	R 236	19900
R 24	1066	R 67	3933	R 110	6633	R 152	8830	R 194	12232	R 237	20202
R 25	1135	R 68	4004	R 111	6700	R 153	8889	R 195	12346	R 238	20513
R 26	1200	R 69	4065	R 112	6768	R 154	8949	R 196	12461	R 239	20833
R 27	1266	R 70	4128	R 113	6838	R 155	9009	R 197	12579	R 240	21164
R 28	1333	R 71	4206	R 114	6908	R 156	9070	R 198	12698	R 241	21505
R 29	1401	R 72	4274	R 115	6981	R 157	9132	R 199	12821	R 242	21858
R 30	1467	R 73	4329	R 116	7055	R 158	9195	R 200	12945	R 243	22222
R 31	1533	R 74	4400	R 117	7130	R 159	9259	R 201	13072	R 244	22599
R 32	1601	R 75	4459	R 118	7207	R 160	9324	R 202	13201	R 245	22989
R 33	1667	R 76	4535	R 119	7246	R 161	9390	R 203	13333	R 246	23392
R 34	1734	R 77	4598	R 120	7286	R 162	9456	R 204	13468	R 247	23810
R 35	1799	R 78	4646	R 121	7326	R 163	9524	R 205	13605	R 248	24242
R 36	1867	R 79	4695	R 122	7366	R 164	9592	R 206	13746	R 249	24691
R 37	1932	R 80	4745	R 123	7407	R 165	9662	R 207	13889	R 250	25157
R 38	1999	R 81	4796	R 124	7449	R 166	9732	R 208	14035	R 251	25641
R 39	2067	R 82	4848	R 125	7491	R 167	9804	R 209	14184	R 252	26144
R 40	2133	R 83	4902	R 126	7533	R 168	9877	R 210	14337	R 253	26667
R 41	2200	R 84	4957	R 127	7576	R 169	9950	R 211	14493	R 254	27211
R 42	2268	R 85	5013					R 212	14652	R 255	27778

CHAPTER 18 - ELECTRICAL SPECIFICATIONS

CY545B Electrical Specifications

Absolute Maximum Ratings

Ambient Temperature under bias ..... 0C to +70C
Storage Temperature ..... -65C to +150C
Voltage on any pin with respect to ground ... -0.5V to Vcc+0.5V
Power dissipation ..... 1.0 Watts

DC & Operating Characteristics

(TA = 0C to +70C, Vcc = +5V +/-10%)

SYM	PARAMETER	MIN	MAX	UNIT	REMARKS
ICC	pwr supply current		26	mA	
VIH	input high level	1.9	Vcc	V	(3.5V for XTAL, RESTART)
VIL	input low level	-0.5	0.9	V	
ILO	data bus leakage		10	uA	high impedance state
VOH	output high level	2.4		V	IOH = -60 uA
VOL	output low level		0.45	V	IOL = 1.6 mA
FCY	crystal frequency	3.5	16	MHz	see clock circuits

Electrical Conventions

All CY545 signals are based on a positive logic convention, with a high voltage representing a "1" and a low voltage representing a "0". Signals that are active low are indicated by a bar over or slash after the pin name, i.e., BUSY/.

All input lines except the data bus include weak pull-up resistors. If the pins are left open, the input signals will be high. The data bus pins must have external pull-up resistors to output a high value. Where appropriate, an input line will be considered in the floating state if the CY545 can drive it both high and low.

The data bus is bi-directional, and is tri-state during nonactive modes. Note that data bus signals are positive logic.

Reset Circuitry

The Restart line (pin#9) must be held high upon power-up to properly initialize the CY545. This may be accomplished by using a 4.7 µFd capacitor, as shown in Figure 18.1. Restart must be high for 10 milliseconds after power stabilizes on power-up. Once the CY545 is running, Restart need only be high for about 10 microseconds (11 MHz crystal).

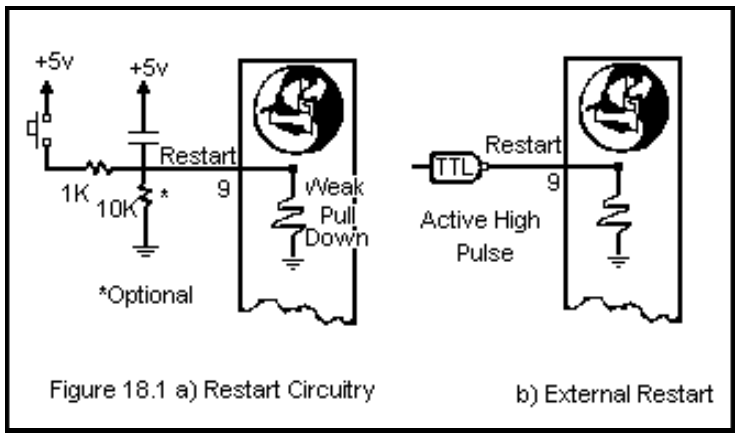


Figure 18.1 a) Restart Circuitry b) External Restart

Clock Circuits

The CY545 may be operated with crystal or external clock circuits. These two options are shown in Figure 18.2. All timing discussed in this manual assumes an 11.059 MHz series resonant crystal. Note that 11.0 MHz, such as a CTS Knights MP110 or equivalent will work fine. The CY545 requires an 11 MHz clock in order to generate the standard baud rates, although any crystal in the allowable range can be used with the adaptive mode, within the timing resolution limits of the CY545. Use 7.3728 MHz to make the fastest possible rate a standard 38400 baud, or 14.7456 MHz for 76,800 baud.

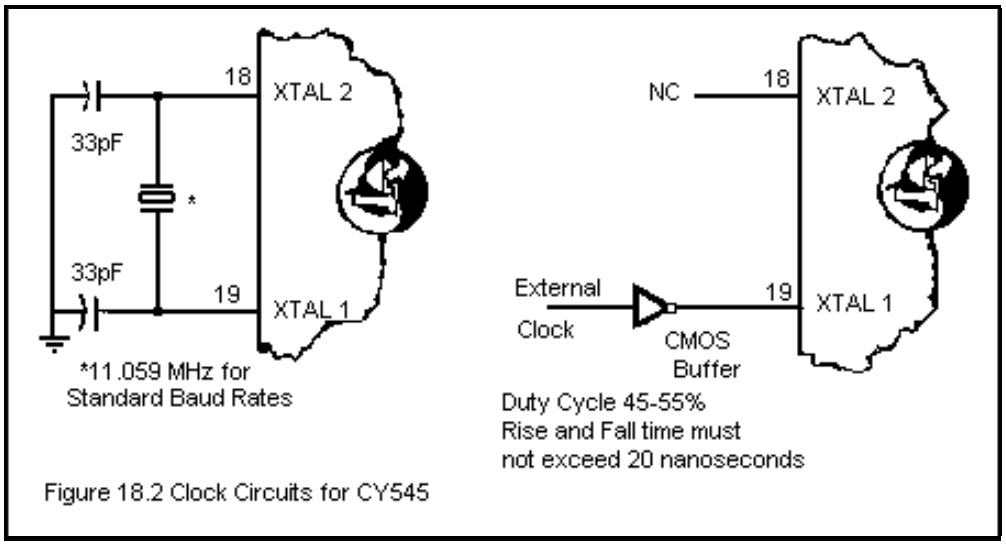


Figure 18.2 Clock Circuits for CY545

CHAPTER 19 - CIRCUITS AND EXAMPLES

Variable Stock Cut and Punch

Suppose we want a row of equally spaced holes punched into some raw stock, which is fed into our machine in continuous strips. The spacing of the holes and the number of holes should be variable for any run of pieces. The stock should be cut into strips with a length that depends on the number of holes and spacing, and the operation should be repeated until the CY545 is signaled to stop.

The capabilities of the CY545 make this problem fairly easy to solve. The command sequence shown below can be used to implement all the functions required. The command sequence can be located in external memory, where the CY545 can directly read and execute the program. The variable information, for the number of holes and spacing between them, can be read from thumbwheel switch inputs. The user selectable function bits act as control and test signals to guide the program. When the operation is complete, the CY545 will stop the program and wait for more commands.

In the following command sequence, memory addresses have been shown for your convenience. They are not sent to the CY545 or recorded in the memory. Counting the bytes as the commands are written makes it easier to determine the addresses for program jumps.

The example program consists of several sections. First, the stepping performance parameters are set, and the current position is cleared to zero. This position could also be found by a Home command, if the application requires a move to a known reference position.

Next, the CY545 waits for a signal that indicates stock is available. This is tested on User Bit 3. When stock is available, the CY545 moves the motor 200 steps from the initial position. Notice that this move is fixed by the command in the program. Here the first move is always 200 steps.

Now the CY545 reads the external thumbwheel switches for the number of steps to move between punching holes. This number replaces the 200 in the Number of Steps Register.

At this point, the CY545 punches the first hole, signaled by User Bit 1. When the bit goes low, the punch is activated, and when it goes high, the punch retracts. Delays are placed around the activation and deactivation to allow for settling time and time to perform the punching operation.

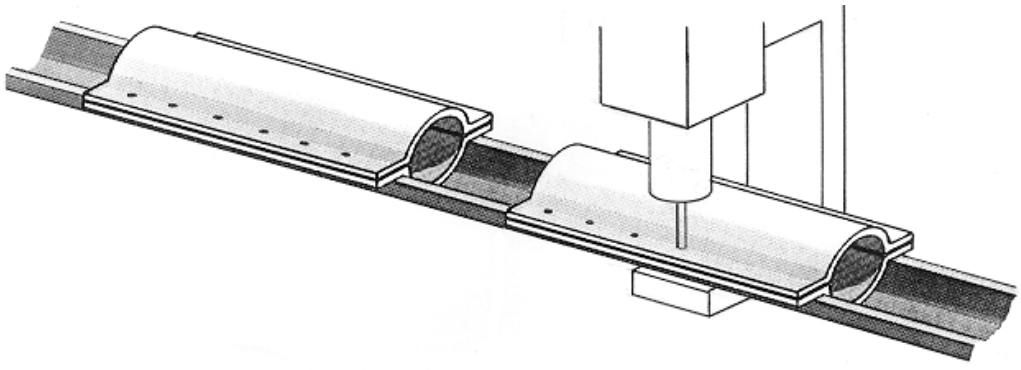
The CY545 continues by moving to the next hole position, as previously set by the N #<cr> command. It then executes the Loop command, which reads the repeat count from external thumbwheel switches, and jumps back to the Delay instruction at the start of the hole punch sequence. This sequence is repeated until the count is exhausted, and the specified number of holes has been punched.

```
Y 100<cr>       Select external memory or location
E<cr>           Write the following commands to memory

100   R 175<cr>     Commands will start at address 100
106   F 12<cr>      First set up stepping parameters
111   S 200<cr>
117   A 0<cr>       Declare current position as zero
121   W 3<cr>       Wait for Bit 3 high, stock is available
125   N 200<cr>
131   +<cr>
133   G<cr>         Move 200 steps before first hole punch
135   N #<cr>       Read hole spacing from external switches
139   D 250<cr>     Delay 250 msec, let stock settle into place
145   /B 1<cr>      Activate punch with Bit 1 low
150   D 1000<cr>    Delay 1 sec to punch hole
157   B 1<cr>       Turn off punch
161   D 100<cr>     Wait for punch to clear work stock
167   G<cr>         Move to next hole position
169   L #,139<cr>   Repeat punches as set by external switches
177   D 250<cr>     Delay 250 msec, let stock settle into place
183   /B 1<cr>      Activate punch for last hole in piece
188   D 1000<cr>    Delay 1 sec to punch hole
195   B 1<cr>       Turn off punch
199   D 100<cr>     Wait for punch to clear work stock
205   N 200<cr>
211   G<cr>         Move 200 steps after last hole punch
213   D 250<cr>     Delay 250 msec, let stock settle into place
219   /B 2<cr>      Cut stock off with Bit 2 low
224   D 1500<cr>    Delay 1.5 sec to cut stock
231   B 2<cr>       Turn off cutter
235   D 150<cr>     Wait for cutter to clear work stock
241   P 0<cr>       Return to starting location
245   T 4,121<cr>   Repeat operation until Bit 4 is high
253   0<cr>         Stop running program now

Q<cr>           Quit saving commands to memory

Y 100<cr>       Reset memory address to start of program
X<cr>           Execute the above program
```



Notice that every character of a command takes one memory location, and all characters must be counted to compute the target address for the Loop command.

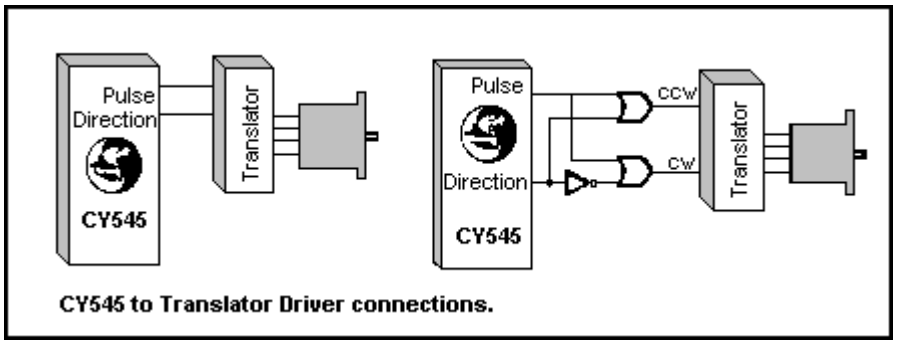
The program then punches a hole at the last position, moves an additional 200 steps, and activates the User Bit 2 signal to cut the strip.

Now the CY545 returns the motor to the starting position. It then tests User Bit 4, and if the signal is low, the program repeats from the Wait instruction. If the signal is high, the program stops with the Zero command, and the CY545 returns to the command mode.

Driver Circuit Considerations

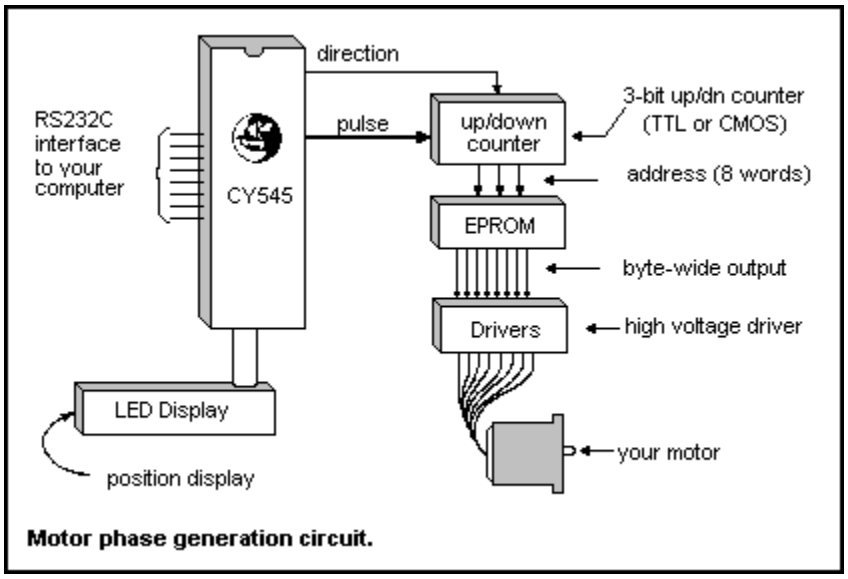
The CY545 provides the timing and logical signals necessary to control a stepper motor. However, to make a complete system, a driver circuit must be added to the CY545. This circuit will take the logical signals generated by the CY545 and translate them into the high-power signals needed to run the motor.

The user has two choices in the selection of driver circuits. Existing designs, usually in the form of pulse-to-step translators, may be used, or special designs may be created. Translators usually require a pulse and direction input, or two pulse streams, one for CW stepping and one for CCW stepping. The translator takes the pulse inputs and generates the proper four-phase outputs for the motor. Note that it is also possible to drive motors with this scheme which are not four-phase designs. Since the translator generates the actual motor driver signals, it only requires the pulse timing and direction information generated by the CY545 PULSE and CCW signals. This allows the CY545 to control three and five-phase motors as well as the standard four-phase designs.



CY545 to Translator Driver connections.

If the user opts for his own driver design, the PULSE and CCW lines may be used to drive a counter circuit, which counts up or down once for each pulse, based on the level of the CCW signal. The counter output then drives the address lines of a memory device, such as a PROM, EPROM, or EEPROM, with data outputs that generate the desired motor phase patterns as the counter address steps through the PROM locations. This design is shown here.



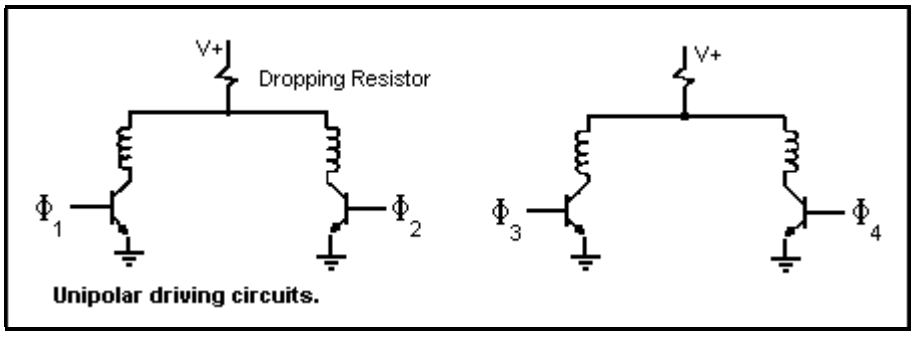
Motor phase generation circuit.

The data outputs from the PROM are then connected to the power driver circuit itself, generating the proper phase patterns for the motor. Note that such a design not only handles motors of various phases, but can also implement stepping schemes such as half-step or quad-step, as well as the standard full-step pattern.

The following paragraphs are meant as a guide to various types of driver circuits, but should not be used as final driver designs. Detailed switching characteristics, transient suppression, and circuit protection logic have been omitted for clarity and simplicity.

Unipolar designs are the simplest drivers, and are generally useful when running at less than 600 steps per second. These designs require motors with six or eight leads, since the power supply is connected to the middle of each winding. The end of each winding is pulled to ground through a transistor controlled by one of the phase output lines from the data PROM.

Motor performance may be improved by adding a dropping resistor between the power supply output and the center tap of each winding. This decreases the field decay time constant of the motor, giving faster step response. The performance increase is paid for by a higher voltage power supply and heat losses through the dropping resistors. This type of circuit is know as an L/xR circuit, where the x represents the resistor value relative to the winding resistance. An L/R circuit would not have any external resistors, while an L/4R circuit would use a resistor of three times the value of the motor winding resistance. Note that the power supply could be four times the nominal motor value with this circuit. Also note that this circuit requires only a single voltage and one transistor per phase.

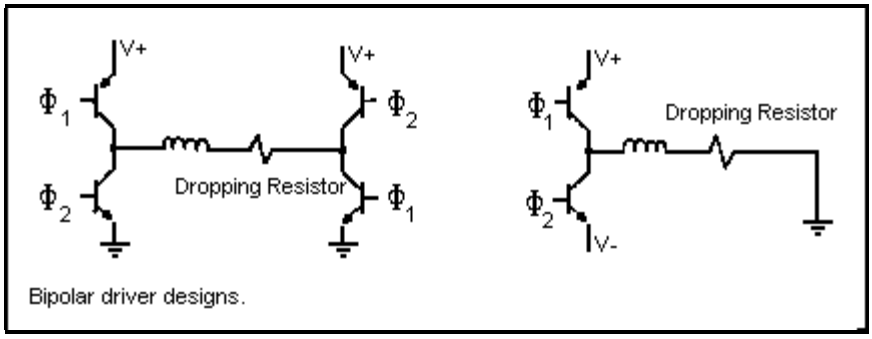


Unipolar driving circuits.

The second basic type of driver circuit is the bipolar design. In this case, the motor is driven only from the ends of each winding, with switching logic used to control the direction of current through the winding. These circuits may be implemented with a four lead motor, since only the ends of each winding are needed.

Bipolar designs are more efficient in driving the motor, and result in higher performance than the unipolar designs. There is also some gain in torque, since the entire winding is always driven, unlike the unipolar design, in which only half of a winding is used at a time.

Two methods of switching the direction of current may be used. With a single voltage power supply, eight transistors are used, two per phase. Transistors are turned on in alternate pairs across each winding to control the current. The second alternative uses only four transistors, but requires a dual voltage power supply. In this case, one side of each winding is connected to ground, and the other side is switched between the positive and negative power supplies. In both designs it is very important to insure that both transistors on one side of the winding are not on at the same time, as this would short the power supply through the transistors, generally destroying the transistors in the process. Protection logic is usually included to insure that one transistor is off before the other is allowed to turn on.



Bipolar driver designs.

The most advanced driver designs are variations on the unipolar or bipolar types, although they are generally implemented using the bipolar approach. These drivers are capable of the highest step rates attainable. They work by switching current or voltage through the motor at much higher than the rated value. This is done for only a short period of time, causing the magnetic field in the motor to change very quickly, without exceeding the maximum power dissipation of the motor. As long as the average dissipation does not exceed the motor rating, the motor will perform without problems. Once the maximum limit is reached, the motor may overheat and self destruct.

One technique for increasing motor performance would simply apply a high voltage to the motor at the beginning of each step. This makes the motor react very quickly to the change in phase signals. After a short period of time, the voltage is switched to a lower value, allowing the motor to continue it's step without overheating.

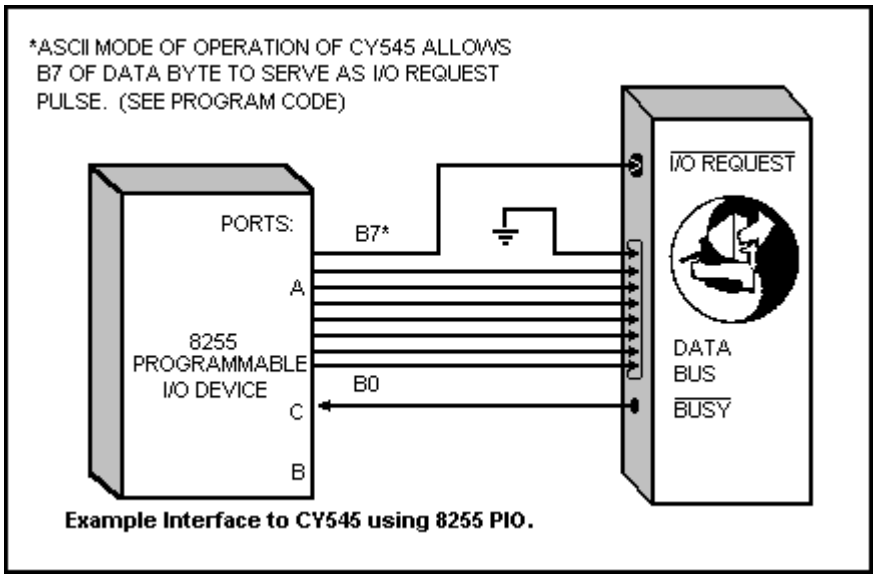
A second approach, known as a constant current design, senses the amount of current flowing through the winding, and adjusts the voltage applied to the motor to maintain the current at its maximum rated value. At the beginning of a motion, the voltage would be low, with a constant adjustment to a higher value as the motor speed increases, and back EMF decreases the current draw for a fixed voltage level.

Another technique, known as chopping, may also be applied to these driver designs. This approach applies a voltage much higher than the rated value for a short period of time. The voltage is then turned off for another time period. This occurs many times per step, with the frequency of switching known as the chopping frequency. This frequency may be controlled by time, switching at a given rate, or it may be controlled by sensing the current flow through the motor, switching at a variable rate. This switching technique controls the power dissipation of the motor, keeping it at maximum torque for any step rate, by adjusting the average applied power. The highest performance drivers are usually designed as bipolar chopper circuits.

The user should consult design guides available from the various motor manufacturers and driver circuit designers for additional information.

Interfacing to the 8255

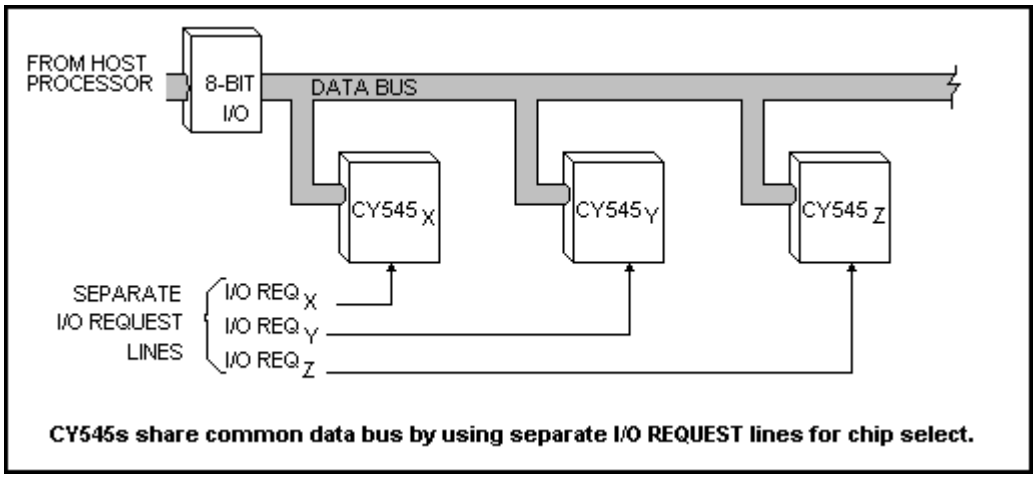
In the example shown in Figure 19.9 below, the CY545 is operating in the parallel ASCII input mode. In this mode, bit 7 is always zero and b7 line of the CY545 data bus may be tied to ground. Since the user will normally transfer bytes of data from memory to the output port, the most significant bit of the data byte may be used to generate the IO_REQUEST strobe, thus allowing only one 8 bit output port to suffice. Of course, a separate port line may be used to generate IO_REQUEST, if desired. If the CY545 is operated in the parallel BINARY mode, all 8 data bus lines are used, and a separate IO_REQUEST line is required. Note that in the example shown, use is made of the fact that the data and the IO_REQUEST signal may be applied simultaneously in parallel operation. If query commands are to be used, all 8 bits of the data bus must be free to operate bidirectionally. In this case, it is generally best to make IO_REQUEST a separate line from the data ports.



Example Interface to CY545 using 8255 PIO.

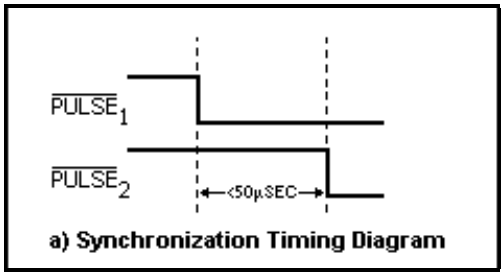
Operating Several CY545s Using a Common Data Bus

In systems where multiple CY545s are to be controlled by a host computer it is possible to use one eight-bit port to establish a common data bus for sending instructions to the CY545s. Each of the separate BUSY lines (pin 15 of each CY545 must be monitored individually and each IO_REQUEST line (pin 13) must be activated separately. This technique effectively uses the IO_REQUEST line as a chip select (CS). A CY545 will ignore all bus information if its IO_REQUEST line is inactive.



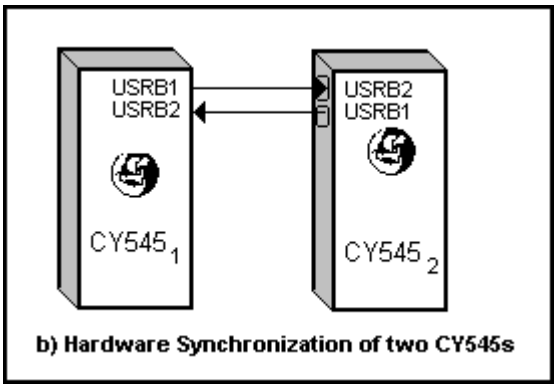
CY545s share common data bus by using separate I/O REQUEST lines for chip select.

Synchronization of Two CY545s



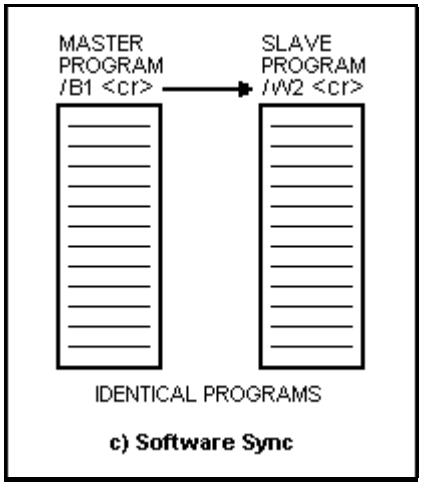
a) Synchronization Timing Diagram

Two CY545s, executing similar programs from external memory, may be synchronized as shown in the figures. The master CY545 can control a USERBIT line of the slave CY545 via the Bit command, to set or clear a user bit. The slave CY545 is started first, with an Execute command, and executes a Wait command and waits until the USERBIT line is driven low by the Bit command executed by the master CY545, when it receives the (second) Execute command. Both CY545s then proceed to the next instruction and are synchronized as shown in the timing diagram, to within several tens of microseconds.



b) Hardware Synchronization of two CY545s

Note that when the two programs are not identical, the master can also use a Wait command, while the slave executes its own Bit instruction, to achieve a more general synchronization.



c) Software Sync

Coordination of Several CY545s

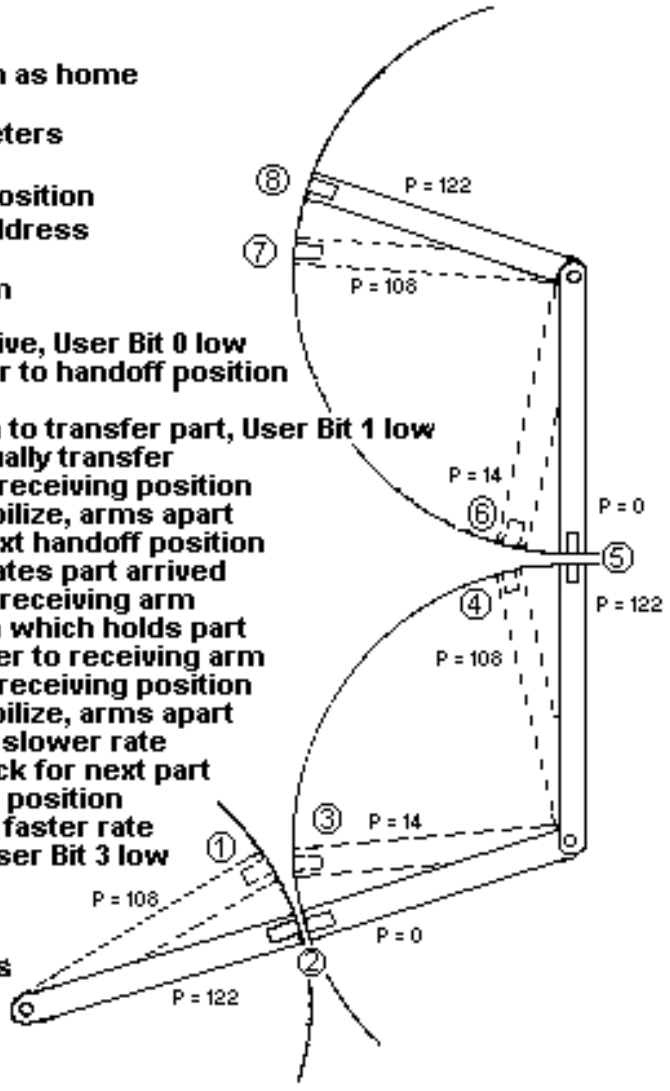
Multiple CY545s may be synchronized to each other by use of the programmable **User Bits**, the **Wait** commands, **Til** commands, and time delays. These may also be combined with other signals, such as CCW, SLEW, or STOPPED, used to select the point in the motion when the signal is presented to the waiting controller. Consider a general parts handling function in which the part must be handed off between two controllers. The geometry of the parts and the arms used to carry the parts requires that the hand off be carefully synchronized between the two controllers. The one to receive the part waits at the receiving position until the CY545 which has the part signals that it has arrived. The two arms then move together in a coordinated motion, reaching a point at which the distance between them is a minimum. The part is exchanged and the arms move apart, again in a coordinated motion. Once a certain position is reached, the arms are free to move independently, and continue with their assigned programs. If the motion is repetitious, both controllers can work with the same program, always being resynchronized at the hand off. The following program illustrates such a motion.

```
A 0<cr>    Declare current position as home
R 120<cr>
S 25<cr>   Define stepping parameters
F 5<cr>
P 14<cr>   Move to the receiving position
Y 50<cr>   Set external memory address

E<cr>      Define hand off program

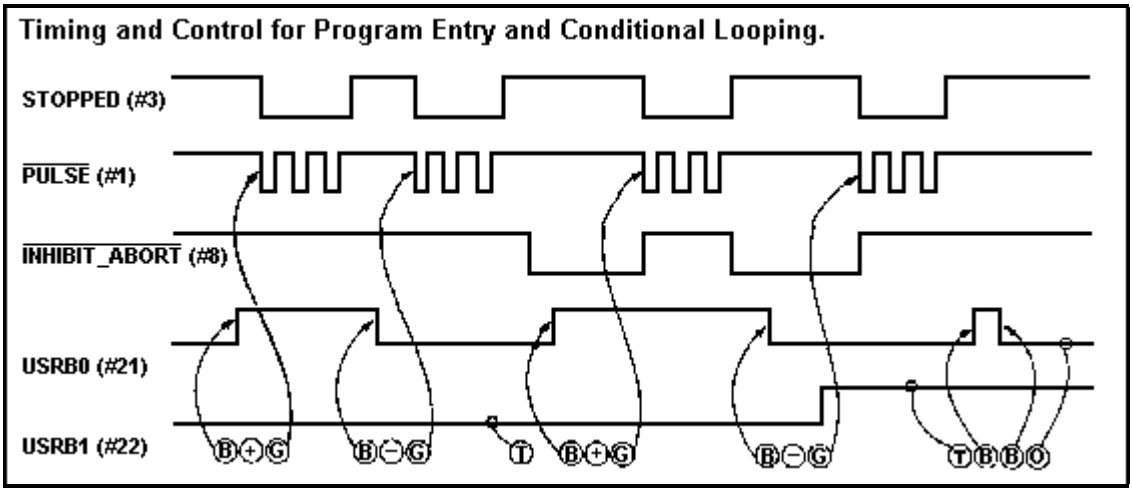
   W 16<cr>  Wait for a part to arrive, User Bit 0 low
   P 0<cr>   Arms move together to handoff position
   +<cr>     Change direction
   B 17<cr>  Activate mechanism to transfer part, User Bit 1 low
   D 90<cr>  Delay for part to actually transfer
   P 14<cr>  Move apart, back to receiving position
   D 90<cr>  Delay for part to stabilize, arms apart
   P 108<cr> Transport part to next handoff position
   B 18<cr>  Low User Bit 2 indicates part arrived
   P 122<cr> Move together with receiving arm
   B 1<cr>   Release mechanism which holds part
   D 90<cr>  Delay for part transfer to receiving arm
   P 108<cr> Move apart, back to receiving position
   D 90<cr>  Delay for part to stabilize, arms apart
   R 20<cr>  Change step rate to slower rate
   P 0<cr>   Move empty arm back for next part
   P 14<cr>  Stay at the receiving position
   R 120<cr> Change rate back to faster rate
   T 3,50<cr>Repeat program if User Bit 3 low
   0<cr>     Else stop program

Q<cr>      End of program
Y 50<cr>   Restart starting address
X<cr>      Execute program
```



Example Programs and Waveforms

The use of the “loop TIL” instruction is illustrated in the following example. The program loops until the USERBIT 1 line (pin #22) goes high, then fetches the next instruction. The effect of the INHIBIT_ABORT input on the STOPPED output is also shown.



PRESET:

B 16<cr>	clear USER BIT O Line
R 90<cr>	set RATE = 90
S 200<cr>	set SLOPE = 200
F 10<cr>	set FIRSTRATE = 10
N 3<cr>	set NUMBER steps = 3
Y 100<cr>	set memory address

ENTER PROG:

	E<cr>	
	B 0<cr>	set USER BIT 0
PROGRAM	+<CR>	set CW direction
CODE	G<cr> GO,	begin stepping
	B 16<cr>	clear output line
	-<cr>	set CCW direction
	G<cr> GO,	begin stepping
	T 1,100<cr>.	repeat above prog Til USER BIT 1 = HI
	B 0<cr>	set USER BIT 0 line
	B 16<cr>	clear USER BIT 0 line
	0<cr>	stop run mode, enter command mode
QUIT:	Q<cr>	
EXECUTE:	Y 100<cr>	reset pointer
	X<cr>	EXECUTE

ASCII and Binary Data Programming Examples

The following command sequence illustrates a simple motion with trigger signals. First we toggle USER BIT 0, then we set the stepping parameters, and finally take five steps. In the ASCII command mode, the command sequence is as follows:

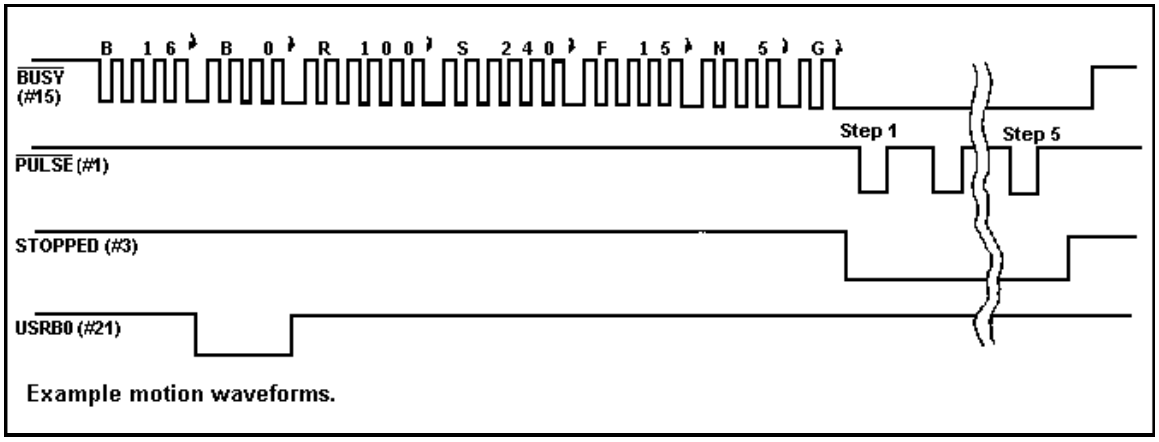
B 16<cr>	USER BIT 0 low
B 0<cr>	USER BIT 0 high
R 100<cr>	set Rate
S 240<cr>	set Slope
F 15<cr>	set First Rate
N 5<cr>	set Number of steps
G<cr>	take five steps

If the same sequence is issued in the Binary command mode, the command letters are the same, but the data values are given to the CY545 in binary form, not as ASCII decimal values. In the Binary command mode, the equivalent sequence is:

42 01 10	USER BIT 0 low
42 01 00	USER BIT 0 high
52 01 64	set Rate 100
53 01 F0	set Slope 240
46 01 0F	set First Rate 15
4E 03 05 00 00	set Number of steps 5
47 00	take five steps

All values are listed as their hexadecimal equivalents, but are sent to the CY545 as single 8-bit data values. Command letters have the same codes as in ASCII, but parameters require a count, followed by the binary parameter value. Notice that multi-byte parameters are sent least significant byte first.

The resulting waveforms are shown below. The CY545 behavior is the same in both ASCII and Binary modes, but the number of data bytes sent to achieve the results is different. Also, Binary mode commands will execute somewhat faster, since the CY545 converts ASCII parameter values into a binary format internally, taking additional command execution time.



Example motion waveforms.

CHAPTER 20 - GETTING YOUR CY545 RUNNING

Getting Your CY545 Running

The following checklist will simplify getting your CY545 up and running.

1. Connect pin 20 to ground and pins 31 & 40 to +5 volts.
2. Be sure RESET (pin 9) is high for at least 10 milliseconds after power stabilizes. The CY545 can be reset at any time.
3. Upon proper reset, all outputs should be at logic 1, and the CY545 should be testing the JOG signal (pin 6).
4. Observe the crystal frequency divided by 6 on ALE, pin 30.

The following steps may be used with the Parallel Command Interface.

5. Place the "+" command on the data bus, ASCII value 2Bh

   D0 = 1
   D1 = 1
   D2 = 0
   D3 = 1
   D4 = 0
   D5 = 1
   D6 = 0
   D7 = 0

6. Lower the IO_REQUEST line, pin 13.
7. Wait for the BUSY line to go low before bringing IO_REQUEST high again. BUSY is pin 15.
8. When IO_REQUEST is brought back high, BUSY will return high.
9. Wait for BUSY to return high, then place the carriage return code on the data bus, value 0Dh

   D0 = 1
   D1 = 0
   D2 = 1
   D3 = 1
   D4 = 0
   D5 = 0
   D6 = 0
   D7 = 0

10. Generate the IO_REQUEST handshake strobe again, interacting with the BUSY signal.
11. Upon completion of the above sequence, CCW (pin 2) will go low.

The following steps may be used with the Serial Command Interface.

12. Connect your serial port to the CY545 RxD and TxD signals, pins 10 and 11, with the proper voltage translation buffers.
13. Send two carriage return codes to adapt the CY545 baud rate. The CRT emulation program at the end of this section is useful for this purpose.
14. Send the command "+<cr>".
15. The CCW signal (pin 2) will go from high to low.
16. Repeat the above command sequence with a "-<cr>" command. The ASCII code for "-" is 2Dh. The CCW signal will return high when this command is sent.

If you have successful parallel or serial communications...

17. If you reach this point successfully, you are commanding the CY545, and should be able to enter any command and obtain the correct response.
18. Suggested sequences:
    a. Try to toggle a User Bit signal using "B 0<cr>" and "B 16<cr>" for pin 21. Remember the single space between the command letter and parameter value.
    b. Take 200 steps with the default parameters, using "G<cr>"
    c. Change the stepping parameters, using the R, F, S, and N commands, then try stepping again.
    d. Refer to examples in this manual for other command sequences.
19. After the initial checkout, use your own command sequences as required by the application. You may need to exercise the external memory interface, thumbwheel switch interface, or an external display.

BASIC Language CRT Emulation Program

The following BASIC language program emulates a CRT for direct serial communications with the CY545A. It starts by sending the two carriage returns to set the CY545 baud rate. It then sends the Mode command to enable the CTS/ signal. These are both done with CTS monitoring disabled. The host serial port id then reconfigured to enable the CTS monitoring function, and all other commands use CTS/. The program is a simple terminal emulator, which takes any command entered from the keyboard and sends it to the CY545, while also displaying any serial data sent by the CY545.

```
100 '                   BASIC Language
110 '                   CRT EMULATOR FOR CY545A
120 '                   Cybernetic Micro Systems, Inc.
130 '
140 '   This program configures the CY545A for serial commands,
150 '   by sending two carriage returns to set the adaptive
160 '   rate, then sending a Mode command to enable the Clear
170 '   To Send signal.  After this, the program redefines the
180 '   computer serial port to function with the CTS signal.
190 '   It then becomes a terminal emulator, waiting for keyboard
200 '   input, which it displays on the screen and sends to the
210 '   CY545A, while displaying all serial characters received
220 '   from the CY545A.
230 '
300 CLS
310 LF$=CHR$(10)  :  CR$=CHR$(13)  : NL$=CHR$(0)  : ES$=CHR$(27)
320 '
330 '   Open the COM1 serial port at 9600 baud, no parity
340 '   Note that CTS is disabled during this part
350 '
360 OPEN "COM1:9600,N,8,1,CS0,DS0,CD0" AS #1
370 '
380 '   Send two carriage returns to adapt the CY545A baud rate
390 '
400 PRINT #1,CR$;CR$
410 '
420 '   Send the Mode command to turn on the CY545A CTS signal
430 '
440 PRINT #1,"O 0A0h";CR$
450 FOR I% = 1 to 5000                  'Delay while Mode Command is sent
460 NEXT I%
470 '
480 '   Now reconfigure COM1 to use CTS
490 '
500 CLOSE #1
510 OPEN "COM1:9600,N,8,1,CS30000,DS0,CD0" AS #1
520 LOCATE 5,5,1
530 PRINT "Ready to Go!"
540 '
550 '   Open the screen for displays
560 '
570 OPEN "SCRN:" FOR OUTPUT AS #2
600 '
610 '   Check for keyboard input.  Display and send any
```

```
620 '   keys to COM port.  Stop when Escape key input
630 '
640 A$=INKEY$  :  IF A$=ES$ GOTO 930
650 IF A$<>" " THEN PRINT #1,A$;  :  PRINT #2,A$;
700 '
710 '
720 '   Check for any received data from COM port, and
730 '   display it, with line feeds filtered out.  Loop back
740 '   to check keyboard again when no more received data
750 '
760 WHILE NOT EOF(1)
770  J%=LOC(1)  :  B$=INPUT$(J%,#1)  :  LF%=0
780  LF%=INSTR(LF%+1,B$,LF$)
790  IF LF%>0 THEN MID$(B$,LF%,1)=NL$  :  GOTO 780
800  PRINT #2,B$;
810 WEND
820 GOTO 640
900 '
910 '   Exit program when Escape key is pressed
920 '
930 CLOSE #1  :  CLOSE #2
940 STOP
```

Changes from CY545 to CY545A

Please note: the CY545 is obsolete.

The CY545 did not have Clear-to-Send ("CTS"). The CY545A implements CTS function on pin 27.

Changes from CY545A to CY545B

Please note: the CY545A is obsolete.

Added Features:

The CY545A did not allow direct commands during program execution, and entering such commands caused command fragmentation, or interleaving of the commands from the two sources. Immediate commands entered at any time during program execution will now be accepted between executing commands and will be correctly interpreted and executed by the CY545B.

The CY545B adaptive baud rate can now adapt up to 57,600 baud (@ 11 MHz). While this rate has always been available as a fixed baud rate, the CY545A was adaptively limited to 300 to 19,200 baud.

The CY545B binary mode now allows ASCII "/" prefix for indicating the complement of a bit function. The CY545A binary mode required the actual complementary numerical values for clearing the bits.

The CY545B changes the 5 millisecond time-out at the end of a motion. The time-out value is now determined by the FirstRate paramenter. Faster rates (higher values) will time out faster than slow rates. This change affects repetitive single stepping, where a single step (N 1) command is embedded in a program loop. The CY545A, because of the fixed time-out could single step continuously at only 200 steps per second (@ 11 MHz), while the CY545B will allow single stepping at up to approximately 1500 steps per second.

The CY545B tests for a free data bus before writing to the HP Display. This will allow slow parallel devices time to get off the bus, and so avoid bus conflicts. The CY545A assumed that parallel devices writing to the bus would do so quickly.

Problems Fixed:

With the CY545A, quoted message strings stored in external memory were not correctly transferred through the parallel interface, and specifically through the parallel interface to the CY233. The CY545B corrects this, allowing successful transfer of parallel quoted strings.

Stepper Motor Controller Selection Guide

CY 500	CY 512	CY 525	CY 545	CY 550	Function
2K	5K	10K	27K	20K	Max Usable Step/Sec
64K	64K	64K	16M	16M	Max Number of Steps
21	25	26	28	38	Number of Instructions
18	48	60	64K	64K	Program Storage (bytes)
Exp	Exp	Lin	Lin	Lin	Accel (Exponent/Linear)
					Command Interface
			●	●	Serial Interface
●	●	●	●	●	Parallel Interface
●	●	●	●	●	Binary Data Structure
●	●	●	●	●	ASCII Data Structure
●	●	●			Internal Stored Program
			●	●	External Stored Program
●	●	●	●	●	Direct Command Mode
			●		Thumbwheel Input Support
			●	●	External Display Support
	o		●	●	Stand Alone Operation
					Program Features
	●	●	●	●	List Prog Buffer Contents
	●	●	●	●	Display # of Steps Param
	●	●	●	●	Display Accel Parameter
	●	●	●	●	Display Step Rate
	●	●	●	●	Display Current Position
			●	●	Display Val of Ext Inputs
				●	Display Motor Status
●	●	●	●	●	Stored Prog Execution
●	●	●	●	●	Conditional Prog Structure
●	●	●	●	●	Programmable Time Delay
	256	256	64K	64K	Program Repetition Count
	●	●	●	●	Unconditional Branch
1	1	1	16	16+	Programmable I/O Lines
				●	Extended I/O
		●			Program Labels
●	●	●	●	●	Multi-controller Sync
		●	●	●	Live Cmds During Prog Exec
				●	Live Cmds while Stepping

● Yes

o To a Degree or with Additional Work

CY 500	CY 512	CY 525	CY 545	CY 550	Function
					Motor Support
	●	●	●	●	Pulse & Direction Output
●	●	●			4-phase Output
●	●	●	o	o	2-phase Compatible
	o	o	o	o	3-phase Compatible
●	●	●	o	o	4-phase Compatible
	o	o	o	o	5-phase Compatible
●	●	●	●	●	Full Step
●	●	o	o	o	Half Step
		o	o	o	Quad Step
			o	o	Micro Step
●	o	o	●	●	Single Step (Jog Mode)
o	o	●	●	●	Continuous Stepping
●	●	●	●	●	Constant Rate Stepping
●	●	●	●	●	Ramped Stepping
			●	●	Low Power Standby
			●	●	Motor On/Off Output
o	o	o	●	●	Limit Detection
					Motion Features
o	o	●	●	●	Programmable Start Rate
●	●	●	●	●	Programmable SlewRate
o	o	●	●	●	Program. Accel/decel Slope
●	●	●	●	●	Software Direction Control
	●	●	●	●	Automatic Direction Finding
●	●	●	●	●	Program. Num of Steps
			●	●	Home Seek Command
●	●	●	●	●	Absolute Posiiton Stepping
●	●	●	●	●	Relative Num of Steps
●	●	●	●	●	Emergency Stop/Abort
	●	●	●	o	Decelerating Stop/Abort
●	●	●	●	●	Step Inhibit Input
	●	o			Closed Loop Rate Control
●			●	●	External Direction Control
●	●	●	●	●	Motion Complete Indicator
	●	●	●	●	SlewIndicator
●	●	●	o	o	Prog Complete Indicator
●			●	●	External Jog Mode
		●		●	On-the-Fly Rate Change
		●		●	On-the-Fly Position Output
				●	On-the-Fly Cmd Execution