interfacing to the TRS-80 computer

MODELS I, III, and 4

TRS-80 TRS-80 TRS-80 TRS-80 TRS-80
TRS-80 TRS-80 TRS-80 TRS-80 TRS-80
TRS-80 TRS-80 TRS-80 TRS-80 TRS-80
TRS-80 TRS-80 TRS-80 TRS-80 TRS-80
TRS-80 TRS-80 TRS-80 TRS-80 TRS-80

Jerry R. Lambert

INTERFACING to the TRS-80 COMPUTER/ Models I, III, and 4

JERRY R. LAMBERT
Clemson University

RESTON PUBLISHING COMPANY, INC.
A Prentice-Hall Company
RESTON, VIRGINIA

Library of Congress Cataloging in Publication Data

Lambert, Jerry R.
Interfacing to the TRS-80 computer/models I, III, and 4.

Includes index.
1. TRS-80 (Computer) 2. Computer interfaces.
I. Title. II. Title: Interfacing to the T.R.S.-80 computer/models I, III, and 4. III. Title: Interfacing to the TRS-eighty computer/models I, III, and 4.
QA76.8.T18L35 1984 001.64 83-9574
ISBN 0-8359-3116-1
ISBN 0-8359-3115-3 (pbk.)

Editorial/production supervision and interior design
by Barbara J. Gardetto

10 9 8 7 6 5 4 3 2 1

Printed in the United States of America

ACKNOWLEDGMENTS

Much of what I know about interfacing I have learned from Dr. Edward W. Page, colleague and friend. Many of the organizational thoughts implemented in this book originated with him. I appreciate the motivation and encouragement that Jane Pardue has contributed. She has also polished and processed the words contained herein.

My students have been patient to point out needed corrections and desirable modifications to improve the material used as class notes. Mike McLeod has provided answers to several technical questions.

AUTHOR'S NOTES

I have attempted to make the hands-on exercises in this book meaningful and as easy as possible to implement. In particular I have tried to choose components that are readily available. Where practical I have used parts available from Radio Shack because of their many stores. To aid you in finding parts to do the exercises, I list them below, including catalog numbers where applicable.

These are available from Radio Shack:

7400 Quad 2-input NAND gate	276-1801
7404 Hex inverter	276-1802
7432 Quad 2-input OR gate	276-1915
7474 Dual D flip-flop	276-1919
74367 Hex buffer (3-state)	276-1835
ADC0804 Analog-to-digital converter	276-1792
LED (typical)	276-026
10 Ω resistor	271-001
220 Ω resistor	271-015
1 KΩ resistor	271-023
4700 Ω resistor	271-030
33 KΩ resistor	271-040
100 KΩ resistor	271-045
TTL relay	275-215
2N2222 NPN transistor	276-1617

DIP switches	275-1301
22 pf ceramic capacitor (2 series 47 pf)	272-121
150 pf ceramic capacitor (100 pf and	272-123
47 pf in parallel)	272-121
0.1 μf ceramic capacitor	272-135
10 μf electrolytic capacitor	272-999
100 μf electrolytic capacitor	272-1016
120 VAC relay	275-217
10 KΩ potentiometer (can also	
substitute for slide)	271-1721
Panel voltmeter	270-1754

Those listed below are available from most electronic supply houses, *e.g.*, Allied Electronics; 401 E. 8th St.; Fort Worth, TX 76102; or Arrow Electronics; 20 Oser Ave.; Hauppauge, NY 11788; or from a mail-order house, *e.g.*, PolyPak.

8212 8-bit latch
7430 8-input NAND gate
8255 24-bit programmable input/output port
5-V dc 1-amp power supply

The General Instrument SP0256-AL2 speech synthesis IC and 3.12 MHz crystal are available from Arrow Electronics. The AD558 digital-to-analog converter is manufactured and sold by Analog Devices; Four Technology Way; Norwood, MA 02062; and is also distributed by Currie, Peak and Frazier, Inc.; P.O. Box 5588; Greensboro, NC 27403.

PREFACE

Most personal computers are used in the manipulation of data, either numbers or characters. Number-crunching programs include accounting packages, spread sheets, and engineering calculations. Word processing, file management, and most games are examples of character handling applications.

A fascinating extension of your computer activities exists in applications to the real world outside your computer. How would you like to have your computer train mice to jump through a hoop, to run a model railroad system from the keyboard, or to serve as a butler at the front door? Your computer can be made to verbally inform you when your garage door is open, to tell you the status of every appliance in your house, or to recommend the type and weight of clothing to wear based on the weather. Even your blood pressure could be measured by your computer. What you can do with your computer in the real world is limited only by your imagination, finances, and time.

Personal computers have tremendous potential in professional use to aid scientific research and for industrial control. Temperature, pressure, weight, flow, sound, light, and distance are just a few examples of the physical variables that can be measured by a data-acquisition system based on a personal computer. Many light industrial processes can be performed by a personal computer. Examples include temperature control, sequencing of machine actions, and lawn irrigation.

Interaction between a computer and the real world requires connection of external devices to the computer. Measurement sensors or control devices cannot be attached directly to the computer, however. Voltage, current, and timing of the devices must all be converted to or from the electrical characteristics of the computer. Conversion is performed by interface circuits. That is what this book is all about—interfacing between the real world and your computer.

Although the TRS-80 computer is emphasized here, the basic principles can be applied to any personal computer you may use. The specific signals and circuits included in this book allow you to see in detail the application of the fundamentals to Model I, III, or 4. If you have a TRS-80 computer, you will be able to build the interface circuits, with parts costing just a few dollars, and see them work. If you have some other computer, you will learn fundamentals which also apply to your computer.

Each chapter of this book is goal-oriented with a specific, stated objective. The fundamentals necessary for the accomplishment of a task are presented and discussed as they apply to the design and construction of interfacing circuits. The information is not intended to cover general interfacing or abstract concepts or to be all encompassing. Rather, it is pointed toward practical education and training to accomplish a specific purpose. Goal-oriented learning is much more efficient!

Chapters 1 through 4 have self-testing built into them so that you can evaluate your understanding of the fundamentals. Chapters 5 through 13 include specific, detailed circuits for you to construct and operate. Several additional projects are also suggested but not detailed. These and others that you can originate make excellent exercises to develop and use for your purpose, whether it be research instrumentation, process control, fair exhibit, or just for fun.

Use of binary numbers is universal in interfacing to any computer. Chapter 1 develops the principles of all modern number systems, including binary, octal, and hexadecimal. Conversion from decimal to binary or from binary to decimal is emphasized because of the necessary, frequent conversions during interfacing circuit design. Chapter 2 continues codes (which is all that a number is anyway), presents BCD and ASCII codes, and defines and demonstrates parity. In Chapter 3 you will learn how a computer is built from an electrical architecture viewpoint. The electronic signals that a computer uses to converse with external equipment are organized into three groups of signals, each with a specific purpose and action. You must know how these signals behave so that you can design circuits to attach to them. A large portion of interfacing consists of digital logic gates. Chapter 4 details the operation of gates and explains how they are combined to form logic circuits. Tri-state logic is introduced in Chapter 4 and implemented in Chapter 5.

Single-bit output to turn on or turn off a single device is developed in

Chapter 5 from the basics of gates, number systems, and computer architecture. The concepts of this chapter must be mastered if the material of succeeding chapters is to be understood. Single-bit input is described in Chapter 6. The hands-on exercise results in a circuit to detect a switch closure, *e.g.*, a microswitch or reed switch to detect an open door. The single difference in doing input operations to the Model III or 4 over a Model I is developed in Chapter 6 also.

Output of eight or more bits simultaneously is described in Chapter 7 using the common 8212 and 8255 integrated circuits. A binary LED display capable of any number between 0 and 255 is described. The display can be used as a clock, counter, or other number indicator. Each of the eight output lines can also be used to control a separate device. Energy management in a house is an example. Input of several bits at a time is described in Chapter 8. Detection of the status of several devices is feasible using 8 to 24 bits to simultaneously read switch closures, etc. Chapter 9 details methods to sense individual input bits and to turn on or turn off individual devices which are all connected to the same input or output port.

A model traffic intersection with automobile detectors embedded in the pavement and signal lights using LEDs is developed in Chapter 10 to demonstrate digital control, using both input and output simultaneously. Hardware interface and software are included to make the TRS-80 speak to you using a speech synthesizer chip. Additional exercises are suggested.

Many process controllers require analog output or input or both rather than or in addition to digital output and input. Chapter 11 describes digital-to-analog converters which are necessary for a digital computer to output an analog signal. What to look for in selecting a D/A converter chip is detailed. A description of the interface necessary for using the popular Analog Devices AD558 is included, enabling you to build the circuit to output a variable electrical signal. Chapter 12 shows how analog-to-digital converters work so that an analog variable such as temperature, pressure, or speed can be input to a TRS-80. The ADC0804 is described and the interface between an incoming analog signal and the computer is developed so that you can build it and see it operate as a position indicator.

Analog control all comes together in Chapter 13. Proportional, integral, and derivative (PID) control, which is so commonly used in industrial controllers, is explained from a beginner's level. A BASIC program to do PID control is developed and implemented in a simulated vehicle steering mechanism.

Chapter 14 is an intensive introduction to the BASIC language used in the TRS-80, intended for those with some previous programming experience. Several example programs and program segments are included.

Again, what you can do by interfacing your computer to the real world is limited only by your imagination and resources. This book may be only the beginning for you, but all the fundamentals are here. Further exercises and activities can be developed by applying your ingenuity and by adapting articles which appear in the various computer magazines.

Happy circuit building!

Jerry Lambert

TABLE OF CONTENTS

chapter 1
NUMBER SYSTEMS

OBJECTIVES: You will learn to convert any number in the decimal, binary, hexadecimal, or octal number system to its equivalent in any other of these four systems.

All communication with and within a microcomputer uses numbers to transfer instructions and data. The number systems used by a microcomputer are not the common decimal system. Therefore, you need to learn to convert numbers from one system to another and to do simple arithmetic in each system.

Early number systems (e.g., the Roman numeral system) formed numbers by repeating basic symbols and adding their values to get a particular number. For example, CXIX = 119. Note that X = 10 regardless of where it appears in the Roman numeral, and that no zero symbol is available.

About 200 B.C., Hindus developed a system based on nine symbols, for the numbers 1 to 9, and based on powers of 10, which they named. For example, "2 sata, 4 dasan, 7" represented 247, and "3 sata, 8" was 308. About A.D. 600 the Hindus invented the sunya (meaning empty), which we call zero, and eliminated place names by substituting the concept of basing values on position. Arabs learned Hindu arithmetic in the 700s, and Europeans began using it around A.D. 1200.

During the late 1600s, the German mathematician Leibniz developed the binary numeration system, but mathematicians found no practical use for the system until computers were developed in the 1940s.

1.1
DECIMAL

The most commonly used number system is the decimal number system. The word *decimal* comes from the Latin word meaning *ten*. Ten symbols are used because people have ten fingers, which we can use for counting. Each symbol is sometimes called a *digit*, from the Latin *digitus*, meaning finger or toe.

The decimal system uses positional notation, whereby the position of each digit in the number determines its value, or weight. For example,

$$825.67$$

is a shorthand notation for

$$(8 \times 10^2) + (2 \times 10^1) + (5 \times 10^0) + (6 \times 10^{-1}) + (7 \times 10^{-2})$$

Note that the position of each digit determines the power of 10 (the *base*, or *radix*, of the decimal number system) by which that digit is multiplied. The products are then added together to get the final number (Figure 1.1). This procedure may seem trivial to you now, but it will be the basis for converting binary and hexadecimal numbers to decimal numbers later in this chapter.

Note three characteristics of all number systems that have positional notation, or use place value:

1. The number of digits is equal to the base, or radix (10 in decimal).
2. The largest digit (9 in decimal) is 1 less than the base.
3. Each digit is multiplied by the base raised to a power that is determined

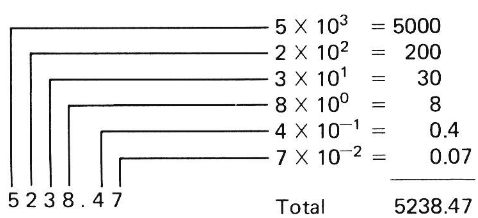

FIGURE 1.1
Calculation of the value of the decimal number 5238.47.