WHAT EVERY ENGINEER SHOULD KNOW ABOUT MICRO-COMPUTERS

HARDWARE/SOFTWARE DESIGN

A STEP-BY-STEP EXAMPLE

William S. Bennett
and
Carl F. Evert, Jr.

WHAT EVERY ENGINEER SHOULD KNOW ABOUT MICROCOMPUTERS

Hardware/Software Design A Step-by-Step Example

William S. Bennett
Link Division
The Singer Company
Binghamton, New York

Carl F. Evert, Jr.
Department of Computer Science
Thomas More College
Fort Mitchell, Kentucky

Illustrated By William S. Bennett
and Joyce M. Beebe

MARCEL DEKKER, INC. New York and Basel

Library of Congress Cataloging in Publication Data

Bennett, William S.
What every engineer should know about microcomputers.

(What every engineer should know ; 3)
Includes index.
1. Microcomputers. I. Evert, Carl F., joint author.
II. Title.
QA76.5.B398 001.64'04 80-12858
ISBN 0-8247-6909-0

MARCEL DEKKER, INC.

270 Madison Avenue, New York, New York 10016

Current printing (last digit):

10 9 8 7 6 5 4 3 2 1

PRINTED IN THE UNITED STATES OF AMERICA

WHAT EVERY ENGINEER SHOULD KNOW ABOUT MICROCOMPUTERS

WHAT EVERY ENGINEER SHOULD KNOW

A Series

Editor

William H. Middendorf

Department of Electrical and Computer Engineering
University of Cincinnati
Cincinnati, Ohio

Vol. 1 What Every Engineer Should Know About Patents, *William G. Konold, Bruce Tittel, Donald F. Frei, and David S. Stallard*

Vol. 2 What Every Engineer Should Know About Product Liability, *James F. Thorpe and William H. Middendorf*

Vol. 3 What Every Engineer Should Know About Microcomputers: Hardware/ Software Design: A Step-by-Step Example, *William S. Bennett and Carl F. Evert,*

Other volumes in preparation

PREFACE

This book is intended to serve the needs of a large class of engineers, managers, sales representatives, technicians, and students who want to gain an insight into the complexities of microcomputer applications, but who do not work with these devices every day. No special knowledge about microcomputers, or even computers, is assumed; technical information is supplied as it is needed throughout the text.

The presentation is built around a simple example, the measurement of the number of gallons of liquid in an oddly-shaped tank, employing a float and a microcomputer. As the solution is developed, facts about microcomputers are brought in, each illustrated by a diagram placed directly next to the idea it illustrates.

The details of the solution are given down to the level of the individual assembly-language instructions and the individual electrical connections, to give the readers a good appreciation for the complexity involved in a microcomputer application.

No attempt is made, however, to treat all of the instruction formats of the selected microprocessor, nor all of the techniques of their use; nor are all of the available auxiliary electronic devices explored. In a similar way, the book does not attempt to cover a range of different microprocessors. The aim is to show newcomers to the field how a

particular problem would be solved, and to impart to them the knowledge they would need in order to solve the problem, thus giving him or her, in a relatively pleasant and painless way, (1) a grounding in a representative set of techniques, (2) a clear idea of what a microcomputer designer or application engineer must go through, (3) an understanding of what must be communicated to a microcomputer development team, should they be in a position to use such services, and (4) enough real understanding of the process so that, if they must obtain for themselves a much larger body of knowledge about microcomputers, they will be better prepared to find, select, and assimilate it.

William S. Bennett
Carl F. Evert, Jr.

ABOUT THE AUTHORS

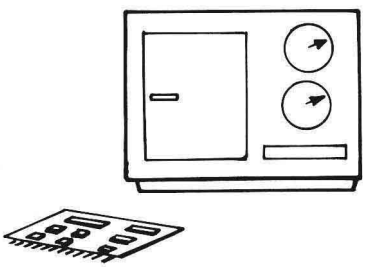
WILLIAM S. BENNETT is a member of the Scientific Staff at the Link Division of the Singer Company. His work involves research and development in computer-generated visual scenes for flight-training simulators which employ a wide range of software and digital hardware. He is the author of some twenty five articles on logic design and microcomputers, including a series of publications on binary logic. That series, over the course of many printings, has introduced an estimated 100,000 engineers to the subject of logic design. Mr. Bennett received his B.S. degree (1952) in electrical engineering from Carnegie-Mellon University.

CARL F. EVERT, Jr. is Chairman and Professor of Computer Science at Thomas More College in Fort Mitchell, Kentucky. He is also Professor Emeritus of Electrical and Computer Engineering at the University of Cincinnati, where he directed a microprocessor application laboratory and taught courses in a variety of computer topics. Professor Evert has taught under UNESCO and NSF sponsorship in India, Romania, and Australia. He received his B.S. and M.S. degrees from Purdue University and his Ph.D. degree from the University of Wisconsin.

CONTENTS

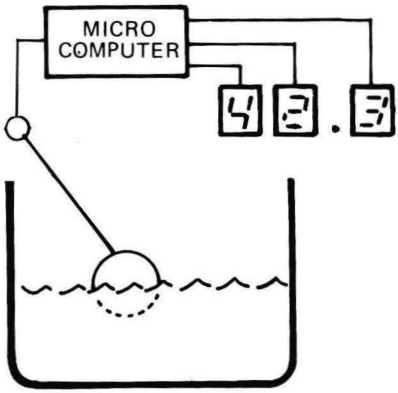
1

INTRODUCTION



In just a few years, the microprocessor industry has grown so much, and the cost of microprocessors has been so drastically reduced, that these devices are now within the reach of every engineer, technician and instrumentation salesman. Almost every new instrument, controller, or data collection system has a microprocessor in it. Engineers of all disciplines, and many other people in technical work, find that they have to have a good understanding of the basic principles of these devices. Fortunately, the field is open to anyone willing to learn a few simple principles, and it is not hard if you spend your time on the right ideas — something this book will help you do. The world of small computers is easy to enter, and once there, it is fun.

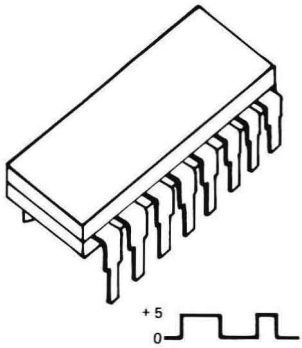
You should think first of the job the microcomputer has to do. We're going to do that in this book; once the job has been established, we'll show you the details of how the microcomputer carries it out. So the design starts by figuring out what the job is — in formal engineering jargon, it starts with an analysis of the system which establishes the requirements, then proceeds to the implementation which accomplishes the result. The steps are easy and logical.



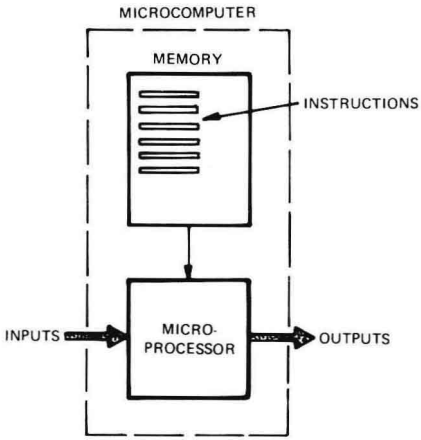
Suppose that the job is to report how much of a certain liquid is in a particular tank, and then display a number or a message that tells how much there is. The tank has an arm with a float on it, which moves up and down as the liquid level moves up and down. What can a microprocessor contribute to the operation of this simple system?

We're going to follow this particular problem in detail, all the way to its solution, in this book. But as we go through the steps of the solution, we'll need to know facts about microcomputers and how they're used; so interspersed with the solution we'll provide

some explanations of those facts, and we can begin with the way a microprocessor chip looks to the outside world:

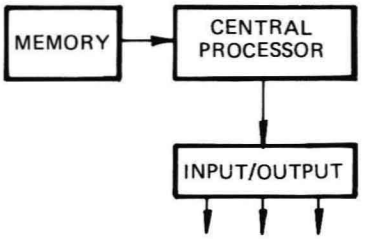


The inputs and outputs of a microprocessor are voltages — either zero or five volts — that appear on several dozen terminals, or "pins," around its outside edge. So if we want to use any sensors as inputs, or displays as outputs, we have to match their voltages, and their arrangements of wires, to the microprocessor in some way.



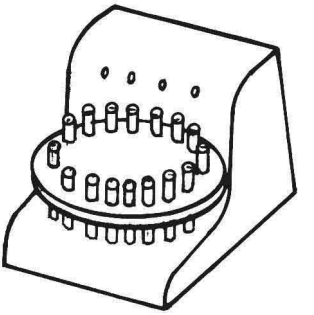
And often there have to be some changes in the input information before it is passed on as an output. Inputs from sensors have to be converted to numbers or alphabetic messages for the output display. The microprocessor can provide an infinite variety of these conversions of input data to output data, but you have to specify exactly what they are. You do it with strings of simple commands, that are put into a permanent memory inside the microcomputer. These strings of simple commands, or instructions, are called programs, and programming is the major activity of thousands of engin-

eers, technicians (and high school students) these days.

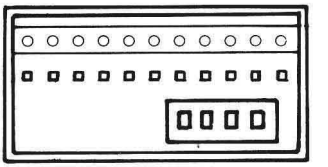


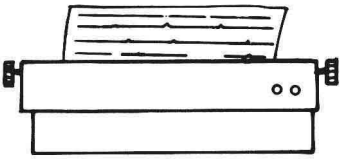
All computers, of course, have a central processor, some memory, and some way to bring data in and present it to the outside world. What is peculiar to the microcomputer is that the processor is usually on one chip, and perhaps memory and input/output circuits are on that same chip, but the memory and I/O circuits are at least in nearby small chips on the same supporting circuit board. Microcomputers are not only small but low in cost, at least relative to the larger computers. Typically, a microcomputer will be used where there is only one or a few different tasks to perform; where the string of instructions, or program, is mostly fixed; where the range of input/output equipment to which it is connected is not as wide as with larger computers; and where speed and efficiency are not as important. A microcomputer, for instance, would be found in a microwave oven, a laboratory instrument, a machine tool, or a cash register terminal: here there is no need for high-speed punched card readers, line printers, large magnetic tape or disk mem-

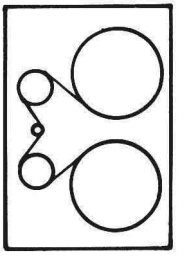
ories, and so forth. The input/output is limited to a keyboard, some digit displays, perhaps a connection through a telephone line to a larger computer somewhere. The program remains the same because the task remains the same: to cook food, to measure blood samples, to guide a cutter, or to compute and print a grocery tape.



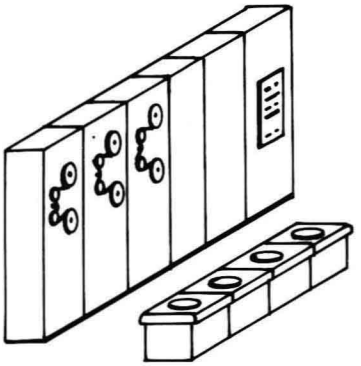
Larger computers, in contrast, handle a generally wider range of jobs. The "minicomputer" is the next step up from a microcomputer; usually it is contained in a box "the size of a breadbox," and has a printing console, perhaps a line printer, and one or more magnetic tape units. It would handle the paperwork for a small office — payroll, inventory, tax computations, sales analyses, and other activities that can change from day to day — though it must be said that the line between microcomputers and minicomputers is blurring, and some well-equipped microcomputers are encroaching on some of these tasks.







Larger still than minicomputers are the "main frame" computers, used, for instance, in large offices,



insurance companies, or the internal revenue service. These computers deal with very large files of data — lists of millions of insurance policies, or tax returns for an entire region of the country. Rows of magnetic disks and tapes will be connected to hold this data, and the input/output problem will be a major part of the task. Other large main frame computers will handle "scientific" computing — perhaps the enormous number of computations needed to produce a daily weather forecast for the nation. There, speed and efficiency in arithmetic, far beyond what is found in the microcomputer, are needed.

If you were to think now just of the microcomputer application — the task that is small, relatively fixed, not too demanding of input/output paths or fast, efficient arithmetic — what would you want in a device to solve the problem? What would you want as the important features of a box that would convert input signals to output information in your device or project or gadget? See if this description doesn't fit what you'd want: