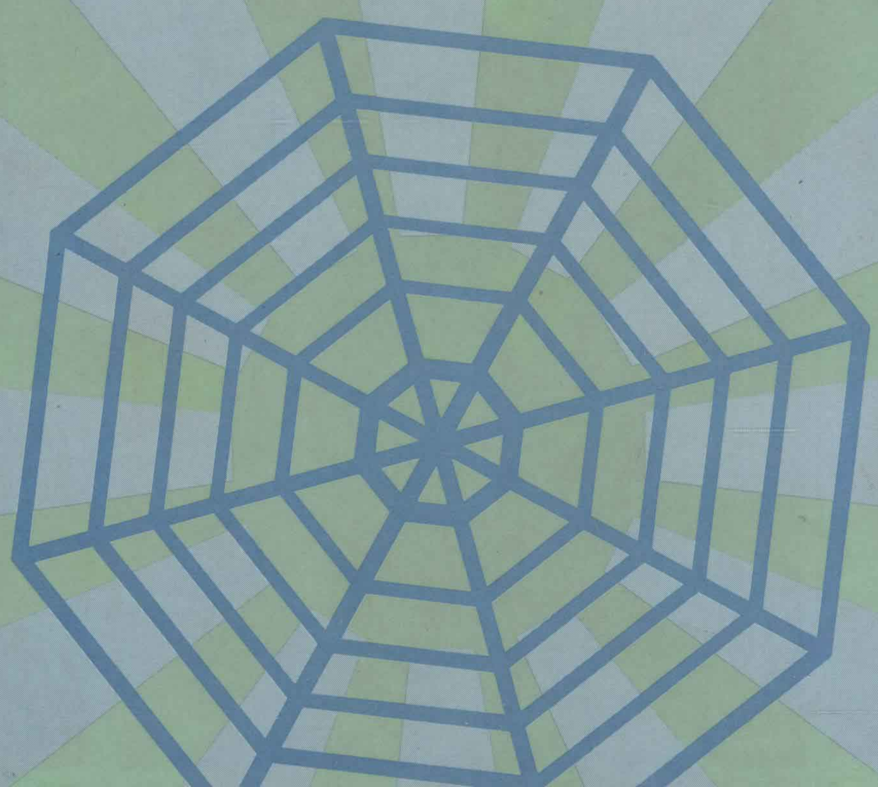
REAL-TIME INTERFACING

Engineering aspects of microprocessor peripheral systems

J.E. COOLING

Van Nostrand Reinhold (UK)



REAL-TIME INTERFACING

Engineering aspects of microprocessor peripheral systems

J.E. COOLING

Department of Electronic and Electrical Engineering
Loughborough University of Technology

VNR UK Van Nostrand Reinhold (UK) Co. Ltd

To Pauline

First published in 1986 by
Van Nostrand Reinhold (UK) Co. Ltd
Molly Millars Lane, Wokingham,
Berkshire, England

Typeset in 10/12 pt Sabon by
Columns Ltd, Reading

Printed in Great Britain by
The Thetford Press Ltd,
Thetford, Norfolk

Library of Congress Cataloging in Publication Data

Cooling, J.E.
Real-time interfacing.

Includes index.
1. Computer interfaces. I. Title.
TK7887.5.C66 1986 621.398 86–9148
ISBN 0–442–31755–7

British Library Cataloguing in Publication Data

Cooling, J.E.
Real-time interfacing: engineering aspects of microprocessor peripheral systems.
1. Engineering — Data processing
2. Microcomputers
I. Title
620'.0028'5416 TA345

ISBN 0–442–31755–7

REAL-TIME INTERFACING

Engineering aspects of microprocessor peripheral systems

Van Nostrand Reinhold
Computing, Electrical and Electronic Engineering

For the complete range of Van Nostrand Reinhold books on these subjects, please send for our catalogue:

Professional & Reference Department
Van Nostrand Reinhold (UK) Co. Ltd
Molly Millars Lane
Wokingham
Berkshire RG11 2PY
England

Preface

This is *not* a book on microprocessors or microcomputers, a subject that is already extensively covered in innumerable text and reference books. It is *not* a theoretical piece of work, but seeks to offer practical solutions to practical problems. Specifically, it looks at the concepts, principles and practices of interfacing microprocessors to plant or machinery systems where reliable and safe real-time operation is required.

Who should read this book?

It is aimed mainly at those who wish to use microcomputers in real-world applications but have little relevant experience. It should be of interest to a wide range of readers from under-graduate project students to engineering managers, via design, development and systems engineers.

How will it be of use to me, the reader?

Most people face two particular problems when they enter a new field of work. First, where do you find relevant and useful information pertaining to that work and, second, having found it, how do you make sense of and assimilate the facts? Microprocessor interfacing is even more of a problem than usual; few books have been written on the topic, most information being contained in data sheets, applications notes and engineering magazines. What I've tried to do here is cover the field of processor interfaces in a comprehensive, clear and logical manner, saving wear, tear, frustration and time for the hard-pressed engineer.

At what level is this book pitched?

I have not made any assumptions about the background of the reader except that he either has, or is in the process of acquiring, a good working knowledge of microprocessors. Some exposure to assembly language software would be useful but is not essential; the same is true of high-level languages. The book is intended to cater for a broad group of engineering disciplines; hence basic concepts that the more experienced reader would take for granted are discussed in detail. And, lest I should be accused of sexism, please accept that 'he' in the text is merely a convenient shorthand notation for 'he-she'.

How should the book be read?

I recommend that you read the opening chapter first, as it has a bearing on the book as a whole. Otherwise individual chapters are self-contained and can be read in any order, except Chapters 8 and 9, which should be taken in sequence. Chapter 10, for the less technically experienced, is best absorbed after Chapter 9.

In general the presentation is aimed at explaining the whys as well as the hows; thus the book is more than a stark presentation of facts.

Are there any special features within this book?

Yes, Chapters 1 and 2. A common reaction after reading a technical book is 'Interesting, but how do I apply these techniques?'. Here, to attempt to relate peripheral interfacing to real-life requirements, a series of practical design problems is discussed throughout the book. To give greater realism to this exercise the interface requirements are defined for a plant controller outlined in Chapter 1. Thus the object of Chapter 1 is to show how in just one plant many different interfaces are needed; it also defines in an engineering manner what the plant is supposed to do and what the requirements of the interfaces are. Each design example is therefore a response to the plant statement of requirements.

Chapter 2, dealing with noise, is really one on its own. Yet it's probably the most important chapter in the book. No matter how clever, advanced or stunning a design is, if it can't withstand the effects of electrical interference you might as well throw it in the dustbin. This particular chapter I would like to dedicate to my ex-colleagues (who must remain anonymous) for their months of site trials in a shipyard trying to stop a new servo-controlled Naval gun nodding in synchronism with the main search radar.

Acknowledgements

I owe a great deal to one person in particular, Alan Cuff, for his support and advice during the writing of this book. Alan, a good friend and ex-colleague from

our days with Marconi, spent many hours over the manuscript; his incisive analysis, combined with doses of critical pessimism, did much to improve the presentation of the final version. I take the blame for the contents.

Many thanks are due to Janet Redman for producing an excellent set of diagrams which are such an important part of the presentation. They represent many woman-hours of work. Finally I'd like to acknowledge the help of my postgraduate student Ghassan Al-Sadiki, for taking the time and effort to produce and test most of the Pascal and ASM86 software.

List of abbreviations

ADC	Analogue to digital converter
ADCCP	Advanced data communications control protocol (ANSI)
AM	Amplitude modulation
AN	Alpha-numeric
ANSI	American National Standards Institute
AO	Analogue output
ASCII	American Standard Code for Information Interchange
BDLC	Burroughs Data Link Control
BG	Bus grant
BIN	Binary
BR	Bus request
BUSAK	Bus acknowledge
BUSRQ	Bus request
CAD	Computer aided design
CB	Complementary binary
CCITT	Consultative Committee on International Telegraph and Telephones
CDCCP	Control Data Communications Control Protocol
CMOS	Complementary metal-oxide-semiconductor
COB	Complementary offset binary
Comms	Communications
CPU	Central processing unit
CRC	Cyclic redundancy code
CRT	Cathode ray tube
CSMA/CD	Carrier sense multiple access/collision detect
DAC	Digital to analogue converter
dB	Decibel
DDCMP	Digital Data Communications Control Protocol
DEC	Digital Equipment Corporation
DIL	Dual-in-line
DIN	Deutsches Institut für Normen
DLCC	Data link control chip
DMA	Direct memory access
DUART	Dual UART
EEPROM	Electrically erasable/programmable read-only memory
EPROM	Electrically programmable read-only memory
EH	Electro-hydraulic
EIA	Engineering Industries Association of America

EL	Electroluminescent
EM	Electromagnetic
EMC	Electromagnetic compatibility
EMI	Electromagnetic interference
EMP	Electromagnetic pulse
EOC	End of conversion
FDM	Frequency division multiplexing
FET	Field effect transistor
FIFO	First-in first-out
FM	Frequency modulation
FSD	Full scale deflection
FSK	Frequency shift keying
GDP	Graphics display processor
HDLC	High level data link control (protocol)
HF	High frequency
HLL	High level language
IEEE	Institution of Electronic and Electrical Engineers
INTAK	Interrupt acknowledge
INTRQ	Interrupt request
I/O	Input/output
ISO	International Standards Organisation
LAN	Local area network
LCD	Liquid crystal display
LED	Light emitting diode
LSB	Least significant bit
LSI	Large scale integration
MDAC	Multiplying digital to analogue converter
mmf	Magneto-motive force
MODEM	Modulator–demodulator
MPSC	Multi-protocol serial communications MSB
MSB	Most significant bit
MUX	Multiplexer
MV	Measured value
NCR	National Cash Register Co.
NRZ	Non-return to zero
NRZ-I	Non-return to zero-inverted
NRZ-L	Non-return to zero-level
OB	Offset binary
PC	Personal computer
PCB	Printed circuit board
PCM	Pulse code modulation
PS	Parallel to serial
PSU	Power supply unit
PWM	Pulse width modulation
RAM	Random access memory
RFI	Radio frequency interference
ROR	Release on request
RS	Recommended standard

RWD	Release when done
SBC	Single board computer
SDLC	Synchronous data link control (protocol)
SH	Sample-hold
SOR	Statement of requirements
SP	Set point
SSR	Solid state relay
TDM	Time division multiplexing
TSBP	Twisted screened balanced pair
TTL	Transistor–transistor logic
UART	Universal asynchronous receiver transmitter
USART	Universal synchronous–asynchronous receiver transmitter
USRT	Universal synchronous receiver transmitter
VDU	Visual display unit
VF	Vacuum fluorescent
VFD	Vacuum fluorescent display
VLSI	Very large scale integration

Contents

1 Peripherals in real-time systems

1.1 INTRODUCTION

The purpose of this work is to describe how peripheral interface systems are used in real-time, real-world microcomputer controllers. Generally these are found within embedded computer systems.

The *Dictionary of Computing* (Oxford Scientific Publications, 1983) gives the following definitions:

Real-time system: any system in which the time at which output is produced is significant. This is usually because the input corresponds to some movement in the physical world, and the output has to relate to that same movement. . . . Examples of real-time systems include process control, embedded computer systems, ...
Embedded computer system: any system that uses a computer as a component, but whose prime function is not that of a computer. One example is that of a weapons-guidance system.

'Embedded computer systems' covers an enormous range of applications, from aerospace fly-by-wire systems to coal cutters in mines. One question should be asked (and answered) right at the outset. What makes these different to traditional computer installations? After all, mainframe designers have been interfacing to keyboards, hard disks, printers, etc., for many years. Surely the underlying principles are the same? Well, yes, but! And it's this but which is crucial, because in real time systems:

(1) The operation is *task* controlled, not processor dictated. If the time requirements of the system cannot be met, then there is no point in even considering a micro-based unit.
(2) Reliability of both hardware and software is a significant factor. Although a failure of the DP computer in the accounts department

may cause a few sleepless nights, it doesn't compare with an avionics failure at a critical point in the flight.

(3) Interfacing to the outside (plant) world is usually the most expensive (and often the most difficult) part of the microcomputer system.

(4) Electrical noise and interference is a major problem in practical environments.

(5) Industrial and military applications impose much higher design and build standards than that normally obtained from computer manufacturers.

These factors, taken together, significantly influence the design of peripheral sub-systems for digital controllers. In the following chapters many facets of this topic are covered. Design examples are given where appropriate, covering both software and hardware aspects. These are considered within the context of an overall control system design for a typical plant application (see Section 1.2).

1.2 THE PLANT — A DESIGN TASK

The plant described here, a fictitious one, is actually based on an experimental system that used electronic analogue controls. It is therefore quite realistic in its requirements and operation. From this starting point the objective is to implement suitable interfaces between the microprocessor controller and the outside world.

Figure 1.1 shows the system under consideration, which is part of a

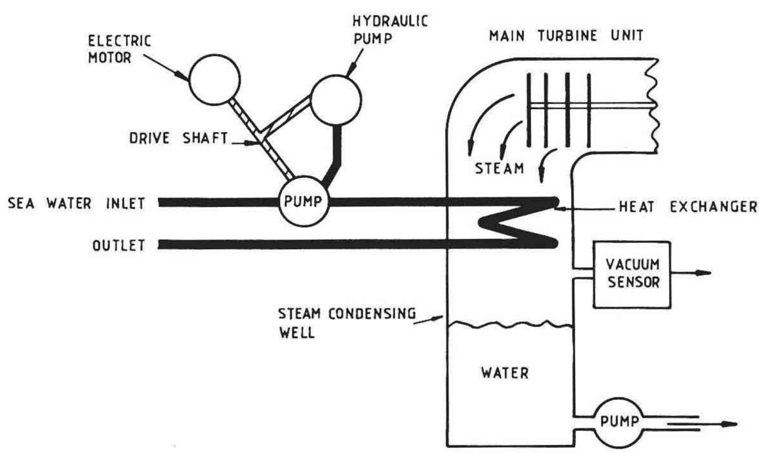


Fig.1.1 Steam-condensing system