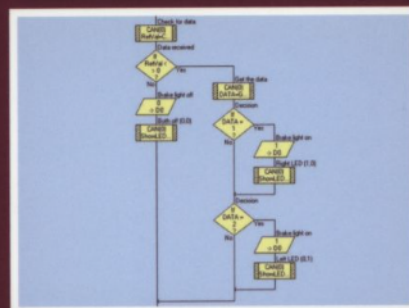
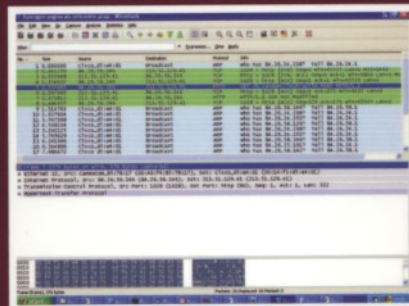


45 projects for PIC, AVR and ARM

elektor

Microcontroller Systems Engineering

45 projects for PIC, AVR and ARM

This book is about a state of the art tool, Flowcode, and how you can use Flowcode to develop microcontroller applications. The book starts very simply with a tutorial project and step-by-step instructions. As you go along the projects increase in difficulty and the new concepts are explained. Each project has a clear description of both hardware and software with pictures and diagrams, which explain not just how things are done but also why. All sources are available for free download.

Since Flowcode is a high level language the intricacies of microcontroller programming are hidden from view. For that reason it doesn't make much difference whether the program is meant for a PIC, AVR or ARM microcontroller. On a high level the programs for these microcontrollers, although vastly different in internal structure, are identical. For that reason this book is on microcontroller application design in general, not just for one type of microcontroller. If you don't own the microcontroller described in a project you can usually convert it to another microcontroller quite easily.

E-blocks will be used as hardware for the projects in this book. This way the hardware can be put together quickly and reliably. Fully tested units simply connect together using connectors or short flat ribbon cables to form completed projects.

This book covers 45 exciting and fun projects for beginners and experts such as:

- timer
- secret doorbell
- cell phone remote control
- youth deterrent
- GPS tracking
- pulse width modulated motor control
- persistence of vision
- sound activated switch
- CAN bus
- Internet webserver.
- and much more

You can use it as a projects book, and build the projects for your own use. Or you can use it as a study guide to learn more about microcontroller systems engineering and the PIC, AVR and ARM microcontrollers.

ISBN 978-0-905705-75-0



Elektor International Media BV
www.elektor.com



Microcontroller Systems Engineering

45 projects for PIC, AVR and ARM

Bert van Dam

Elektor International Media BV

Postbus 11

6114 ZG Susteren

The Netherlands

All rights reserved. No part of this book may be reproduced in any material form, including photocopying, or storing in any medium by electronic means and whether or not transiently or incidentally to some other use of this publication, without the written permission of the copyright holder except in accordance with the provisions of the Copyright, Designs and Patents Act 1988 or under the terms of a licence issued by the Copyright Licensing Agency Ltd, 90 Tottenham Court Road, London, England W1P 9HE. Applications for the copyright holder's written permission to reproduce any part of this publication should be addressed to the publishers.

The publishers have used their best efforts in ensuring the correctness of the information contained in this book. They do not assume, and hereby disclaim, any liability to any party for any loss or damage caused by errors or omissions in this book, whether such errors or omissions result from negligence, accident or any other cause.

British Library Cataloguing in Publication Data

A catalogue record for this book is available from the British Library

ISBN 978-0-905705-75-0

Prepress production: Autronic, Blaricum

First published in the United Kingdom 2008

Printed in the Netherlands by Wilco, Amersfoort

© Elektor International Media BV 2008

Content

	Prologue	9
1	What you will need	12
2	Tutorial	17
2.1	Concept	17
2.2	Hardware	17
2.3	Software	22
2.4	Simulation	29
2.5	Downloading	31
2.6	Operational	32
3	Basic	33
3.1	LCD display	33
3.2	Running light	39
3.3	Secret doorbell	45
3.4	Serial communication	51
3.4.1	RS232	51
3.4.2	USB	58
3.5	Analog to Digital Conversion	62
3.6	Dark activated switch	66
3.7	Youth deterrent	70
3.8	Sound activated switch	77
3.9	Air to fuel ratio	80
3.10	Digital clock	87
3.11	Debugging	91
3.11.1	Simulation	93
3.11.2	Serial connection	94
3.12	Bootloader	96
4	Advanced	100
4.1	Vref+	100
4.2	Record short sounds	105
4.3	Cricket	111
4.3.1	High quality version	111
4.3.2	Low cost version	116
4.4	Custom Characters	119
4.5	Animation	123

4.6	Random	124
4.7	A fan on PWM	128
4.8	Persistence of vision	135
4.9	Flash memory	139
4.10	USART connection	144
4.11	GPS tracking	149
4.12	Use PPP to access EEPROM	155
4.13	Poetry box	159
4.14	Voice command	163
5	Sensors	169
5.1	Infrared object detection	169
5.2	Photometer	172
5.3	Ultrasonic range finder	178
5.4	Digital thermometer	184
5.5	Data sampling the fridge	189
5.6	Heat loss in a residential building	197
5.7	Capture sound frequency	202
5.8	Tow away alarm	212
6	Cell phone	217
6.1	Send a text message	217
6.2	Receive a text message	221
6.3	Remote control	225
7	CAN bus	232
7.1	Remote LEDs	232
7.2	Monitor	240
8	Internet	248
8.1	Ping (Are you there?)	248
8.2	Webserver	258
8.3	Send an alarm e-mail	265
8.4	UDP remote control	276
9	Design your own E-block	282
9.1	Analog test signal	282
9.2	Microphone pre-amplifier	284
10	Migration between PIC, AVR and ARM	288
10.1	Software	288
10.2	Hardware	289
10.3	Example	293

11	Going into production	296
11.1	Youth deterrent (continued)	296
12	Appendix	302
12.1	ASCII table	302
12.2	Visual Basic communications	303
12.3	Tips and tricks	305
12.4	Microcontroller Mathematics	314
12.5	E-block connections	320
	Index	327

Prologue

*You have to do what others won't,
to achieve what others don't.
(Anonymous)*

When I was young my dad taught me how to program a large mainframe computer. First you need to think through what you want the program to do. Then draw a flowchart, which is a graphical representation of the program flow. It consists of symbols connected by lines with arrows. The program will flow from symbol to symbol following the arrows. Each symbol of the flowchart has a specific meaning. The rectangle for example is an operation, the diamond is a decision, the circle a connector to a different part of the program.

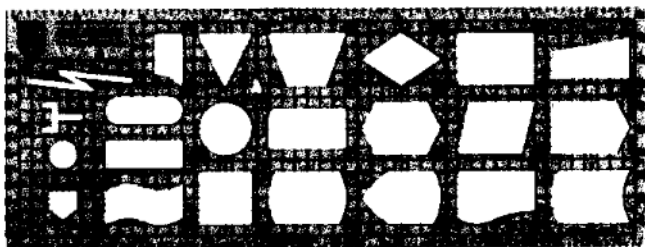


Figure 1. Flowchart template.

Flowcharting is an ideal way to develop a program, because it allows anyone, not just programmers, to understand what the program will do. Once everyone involved is satisfied that this is indeed the program they want, the flowchart must be transformed into lines of code. Then each line is copied onto a cardboard card with a kind of typewriter that punches holes in the card. The stack of cards is fed into a punchcardreader because that was the only way to get the program into the computer. The computer itself was in a different room, in fact it *filled* an entire room, and controlled by people in white lab coats. No one was allowed to get near it, let alone touch it. The results of the program would be printed on large sheets of tractorfed paper. And often there would only be one sheet with one line, such as: Error in line 2816, execution aborted.

We have come a long way. State of the art tools allow you to draw a flowchart directly on your computer screen, and enter in each symbol details of what you want the program to do at that location. With a press on a button the program is simulated allowing for very easy debugging. And with another button the whole program is compiled and downloaded into a microcontroller, a computer the size of a fingernail. You can get a program up and running in a matter of minutes.

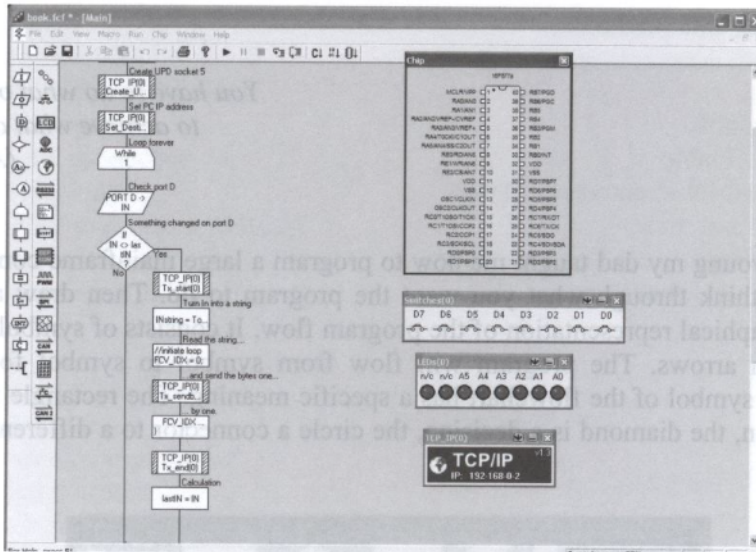


Figure 2. Flowcode state of the art software design.

This book is about such a state of the art tool, Flowcode[®], and how you can use Flowcode to develop microcontroller applications. The book starts very simply with a tutorial project and step-by-step instructions. As you go along the projects increase in difficulty and only the new concepts are explained. Each project has a clear description of both hardware and software with pictures and diagrams, which explain not just how things are done but also why. All sources are available for free download, including the support software¹.

E-blocks[®] will be used as hardware for the projects in this book. This way hardware can be put together quickly and reliably. Fully tested units simply connect together using connectors or short flat ribbon cables to form completed projects. No soldering is required.

Since Flowcode is a high level language the intricacies of microcontroller programming are hidden from view. For that reason it doesn't make much difference whether the program is meant for a PIC, AVR or ARM microcontroller. On a high level the programs for these microcontrollers, although vastly different in internal structure, are identical. For that reason this book is on microcontroller systems engineering in general, not just for one type of microcontroller. If you don't own the microcontroller described in a project you can usually convert it to another microcontroller quite easily, as explained in chapter 10.

¹ Flowcode is not included and must be purchased separately.

This book covers a series of exciting and fun projects such as a secret doorbell, a youth deterrent, GPS tracking, cell phone remote control, persistence of vision and an Internet webserver. You can use it as a projects book, and build the projects for your own use. Or you can use it as a study guide to learn more about Flowcode systems engineering and the PIC, AVR and ARM microcontrollers.

I would like to thank Ben Rowland, Sean King, Steve Tandy and John Dobson for their help and support while writing this book, and Alan Dobson for his help with the final editing.

Bert van Dam
Roosendaal, 2008