
Starting Electronics Microcontroller Basics with PIC



By Tam Hanna



elektor

LEARN > DESIGN > SHARE

Chapter 1 • Introduction and motivation	9
Instead of an introduction – on buying an NE555	9
Why PIC? Why assembler?	10
What do we need?	11
Installing MPLAB on Windows	11
Chapter 2 • Executing Assembly in Sequence.....	15
Getting closer to the PIC.....	15
Building a testing rig.....	18
An introduction to Assembly.....	21
Understanding start-up	23
Dynamic hardware configuration.....	25
What happens in the background?	28
How outputs are configured	30
Conclusion	32
Chapter 3 • Program Sequence Control.....	33
Jump Operators and Pipelining	34
Slow down program execution with NOP	36
Aside: Comments	38
Declaration of variables	39
Conditional jump operations.....	42
Troubleshooting.....	47
Optimizing the program.....	49
Using Subroutines.....	51
The stack and its pitfalls	54
What happens next?	58
Chapter 4 • PIC Arithmetic Operations.....	59
Understanding binary	59
The traps of adding things	61
Comparing numerical values	64
Two's complement	72
Logic operations	76
Shift!	79

Bottom line	82
Chapter 5 • Interrupts and Sleep Modes	83
Understanding interrupt logic	83
Unsharp without interrupts	85
Improving the reaction time	91
Let's save power.....	96
Monitoring with WDT	98
Outlook: timers allsorts	102
Chapter 6 • Indirect Addressing	104
Discrete and continuous	104
Configure the DAC	105
On lookup tables	110
Tables from data memory	112
On disassembly and constants.....	114
Return to table access	117
Tables from RAM	120
Troubleshooting with Assembler.....	125
Jumping across banks	132
Bottom line	132
Chapter 7 • Getting Started With C	134
The C workflow	134
Install MCC	137
Start MCC.....	138
Flashing _a_ LED with MCC.....	141
Analyze code structure	144
Flashing _a_ LED.....	150
Analyzing compiler output.....	151
Bottom line.....	157
Chapter 8 • Interrupts and Timers à la C.....	158
Interrupt handling with C.....	158
Provision of interrupt logic	161
Merging changes (against MCC-generated code)	163

Comparative analysis of execution speed	165
Optimizing C code.....	171
An overview of timers	173
More timer play	184
Using timers to control program execution	186
Good night!	193
Pulsewidth modulation.....	195
Frequency-derived gimmicks	204
Conclusion and outlook.....	207
Chapter 9 • Hardware Accelerated Buses: I²C, SPI and more	208
SPI – if speed is needed	210
Programming SPI	221
I ² C — a bus for many.....	226
I ² C, reloaded	233
Evaluate serial data	247
Bit banging: Wiegand Protocol.....	257
Bit banging: state machine	260
Bit banging: coding.....	261
Bottom line.....	267
Chapter 10 • Storing Data During Power-off	268
Writing to EEPROM.....	268
A matter of speed.....	277
Addressing games	285
HEF, or abuse the program memory	288
Aside: HEF, reloaded	295
The bottom lines	300
Appendix A: Search Ranking.....	301
Appendix B: Fast Spectral Components	303
On waves and parts	304
Deeper analysis!.....	306
What's in it for me?	309

Where do we go from here? 310

 Get help 310

 Ideas from third parties 312

 Contacting the author 314

Index 315

W

Watchdog timer	98
WDTE	99
Wiegand Protocol	257
working register	29
WriteBlock	290
WriteWord	290

X

XOR	76
XORLW	78
XORWF	78