
Programming the Finite State Machine with 8-Bit PICs in Assembly and C



Andrew Pratt



an Elektor Publication

LEARN > DESIGN > SHARE

Table of Contents

• Preface	9
Chapter 1 • Getting Started	10
1.1 • Introduction.	10
1.2 • Practical Implementation	11
1.2.1 • Choice of Operating System	11
1.2.2 • Machine Language and Assembly Source Files	11
1.2.3 • Using the Applications on Microsoft Windows	13
1.2.4 • Installing the Applications on Linux	14
1.2.5 • The FTDI Lead and Testing the Programming Chain.	15
1.3 • Some Fundamentals	18
1.3.1 • Bits and Bytes	18
1.3.2 • The Hexadecimal Numbering System	20
1.3.3 • Boolean Logic	21
1.3.4 • Bitwise logic	21
1.3.5 • PIC Architecture	22
1.3.6 • Data Memory Organization	22
1.3.7 • An Assembly Program Snippet	23
1.4 • Program Memory	26
1.5 • Hello World	27
1.5.1 • Some Key Points	27
1.5.2 • Assembling the Program.	29
Chapter 2 • The Assembly Program as a Finite State Machine	32
2.1 • Introduction	32
2.2 • A State Machine Framework for Assembly Language.	32
2.3 • Timer0	33
2.4 • Interrupts	35
2.5 • A More Complicated LED Flasher	41
2.6 • Running More Than One Machine in a Program	43
2.7 • Driving a Seven Segment LED Display	45
2.8 • The Differences Between the PIC 12F1822 and the 16F1823	51
2.9 • Interrupts and State Diagrams	52
Chapter 3 • Macros, Subroutines and Bank Switching	53
3.1 • Introduction	53
3.2 • Create Your Own Instruction	54

3.2.1 • Use an Include File	55
3.2.2 • More Macros.	56
3.2.3 • Conditional Assembly.	60
3.3 • Subroutines.	61
3.3.1 • An Example of a Subroutine	61
3.3.2 • Return Address and the Stack	63
3.3.3 • Calculating the Delay	63
3.3.4 • Calling Subroutines from Subroutines	64
3.4 • Bank Switching	65
Chapter 4 • Inputs and Outputs	67
4.1 • Introduction	67
4.2 • Serial Output to a Computer.	67
4.2.1 • TTL Level Serial Communications.	67
4.2.2 • Configuring the EUSART to Transmit a Byte	68
4.2.3 • Serial Output Example	69
4.3 • Serial Input from a Computer	72
4.3.1 • Configuring the EUSART to Receive Bytes	73
4.3.2 • Interrupt Service Routine	74
4.3.3 • Serial Input Example	75
4.4 • Analog Inputs	76
4.4.1 • Setting ADCON0 and ADCON1.	78
4.4.2 • Circuit and State Diagram	79
4.5 • Pulse with modulated outputs.	81
4.5.1 • LED with Pulsing Brightness	82
4.6 • Digital Inputs.	85
4.6.1 • Counting Input Pulses From a Switch	85
4.6.2 • First method eliminating the effect of switch bounce	88
4.6.3 • A Better Method of Eliminating the Effect of Switch Bounce	90
Chapter 5 • Project Hardware Construction	93
5.1 • Introduction	93
5.2 • Overview of the Suggested Method	93
5.3 • Cutting and Drilling the board.	94
5.4 • Populating and Wiring the Board	96
5.5 • The Circuit Board Test Program.	97
5.5.1 • Analog Configuration	99

Chapter 6 • Binary Arithmetic	103
6.1 • Introduction	103
6.2 • Binary Addition of Unsigned Numbers	103
6.2.1 • Adding Two Eight Bit Positive Integers	104
6.2.2 • Serial Read Program Command Line Arguments	107
6.2.3 • Adding Two Sixteen Bit Positive Numbers	108
6.3 • Binary Subtraction of unsigned integers	111
6.4 • Binary Subtraction with Negative Results	113
6.5 • Negative numbers in binary	114
6.6 • Binary Multiplication	116
6.7 • Binary Division	119
Chapter 7 • Digital Voltmeter Project	126
7.1 • Introduction	126
7.2 • The State Diagrams	126
7.3 • Scaling the Raw Analog Value	133
7.4 • Extracting the individual figures for the display	135
7.5 • Detecting No Input Voltage	135
7.6 • Recalibration	135
Chapter 8 • Troubleshooting and Planning	136
8.1 • Introduction	136
8.2 • Have an Overview of the Project	136
8.2.1 • State Diagrams and Flow Charts	136
8.3 • Break Big Problems Down Into Smaller Ones	137
8.4 • Read Through Your Code in Detail and Add Comments	137
8.5 • Debugging a Running Program	137
8.6 • Traffic Lights	138
8.6.1 • A Circuit Diagram	140
8.6.2 • Separate Different Problems	140
8.6.3 • Producing the Code	141
8.7 • Using Debug Macro on the Voltmeter Programmable	148
8.8 • A List of Things to Remember	150
Chapter 9 • A Comparison with C	151
9.1 • Introduction	151
9.2 • The Microchip XC8 Compiler	151
9.2.1 • XC8 for Microsoft Windows	151
9.2.2 • XC8 for 32 Bit Debian-Based Distributions	151
9.2.3 • XC8 for 64 Bit Debian-Based Distributions	152
9.2.4 • XC8 for 64-Bit Fedora	152
9.3 • Introduction to C	153

9.3.1 • Hello World in C	154
9.3.2 • Using the XC8 Compiler in Microsoft Windows	156
9.3.3 • Using the XC8 Compiler in Linux	157
9.3.4 • Emitted Code Assembly vs C	157
9.3.5 • Interrupts in C	159
9.3.6 • The <code>__delay()</code> functions	161
9.3.7 • Extending the If Statement.	162
9.3.8 • The Switch Statement	163
9.3.9 • An Experiment to Measure Code Speed	167
9.4 • Serial Communication	168
9.4.1 • Serial Byte Transmission	168
9.4.2 • Serial Byte Reception.	169
Chapter 10 • Further C	171
10.1 • Introduction.	171
10.2 • Data Types	171
10.3 • More on Functions.	172
10.4 • Integer Arithmetic.	173
10.4.1 • Transmitting a Four-Byte Integer	173
10.4.2 • The "for" loop	175
10.5 • The Voltmeter in C	179
10.5.1 • Turning Bits On and Off Using Bit Type	179
10.5.2 • Turning Bits On and Off Using Bitwise Operators	183
10.5.3 • Turning Bits On and Off Using Bitfields	183
10.6 • Summary of Assembly, C, and Finite State Machines	184
• Index.	185