

# EMBEDDED SYSTEMS:

---

## INTRODUCTION TO THE MSP432 MICROCONTROLLER

Volume 1

*First Edition*

*January 2019*

Jonathan W. Valvano

*First edition  
6<sup>th</sup> printing  
January 2019*

*The true engineering experience occurs not with your eyes and ears, but rather with your fingers and elbows. In other words, engineering education does not happen by listening in class or reading a book; rather it happens by designing under the watchful eyes of a patient mentor. So, go build something today, then show it to someone you respect!*

ARM and uVision are registered trademarks of ARM Limited.

Cortex and Keil are trademarks of ARM Limited.

Code Composer Studio is a trademark of Texas Instruments.

All other product or service names mentioned herein are the trademarks of their respective owners.

In order to reduce costs, this college textbook has been self-published. For more information about my classes, my research, and my books, see <http://users.ece.utexas.edu/~valvano/>

For corrections and comments, please contact me at: [valvano@mail.utexas.edu](mailto:valvano@mail.utexas.edu). Please cite this book as: J. W. Valvano, Embedded Systems: Introduction to the MSP432 Microcontroller, Volume 1, <http://users.ece.utexas.edu/~valvano/>, ISBN: 978-1512185676.

Copyright © 2019 Jonathan W. Valvano

All rights reserved. No part of this work covered by the copyright herein may be reproduced, transmitted, stored, or used in any form or by any means graphic, electronic, or mechanical, including but not limited to photocopying, recording, scanning, digitizing, taping, web distribution, information networks, or information storage and retrieval, except as permitted under Section 107 or 108 of the 1976 United States Copyright Act, without the prior written permission of the publisher.

ISBN-13: 978-1512185676

ISBN-10: 1512185671

## Table of Contents

<b>Preface</b> .....	<b>x</b>
<b>Acknowledgements</b> .....	<b>xi</b>
<b>1. Introduction to Computers and Electronics</b> .....	<b>1</b>
<b>1.1. Review of Electronics</b> .....	<b>2</b>
<b>1.2. Binary Information Implemented with MOS transistors</b> .....	<b>7</b>
<b>1.3. Digital Logic</b> .....	<b>8</b>
<b>1.4. Digital Information Stored in Memory</b> .....	<b>16</b>
<b>1.5. Numbers</b> .....	<b>17</b>
<b>1.6. Character information</b> .....	<b>29</b>
<b>1.7. Computer Architecture</b> .....	<b>30</b>
<b>1.8. Flowcharts and Structured Programming</b> .....	<b>33</b>
<b>1.9. Concurrent and Parallel Programming</b> .....	<b>37</b>
<b>1.10. Exercises</b> .....	<b>40</b>
<b>2. Introduction to Embedded Systems</b> .....	<b>47</b>
<b>2.1. Embedded Systems</b> .....	<b>48</b>
<b>2.2. Applications Involving Embedded Systems</b> .....	<b>50</b>
<b>2.3. Product Life Cycle</b> .....	<b>53</b>
<b>2.4. Successive Refinement</b> .....	<b>57</b>
<b>2.5. Quality Design</b> .....	<b>60</b>
2.5.1. Quantitative Performance Measurements .....	60
2.5.2. Qualitative Performance Measurements .....	61
2.5.3. Attitude .....	61
<b>2.6. Debugging Theory</b> .....	<b>63</b>
<b>2.7. Switch and LED Interfaces</b> .....	<b>66</b>
<b>2.8. Introduction to C</b> .....	<b>69</b>
<b>2.9. Exercises</b> .....	<b>79</b>
<b>3. Introduction to the ARM® Cortex™-M Processor</b> .....	<b>81</b>
<b>3.1. Cortex™-M Architecture</b> .....	<b>82</b>

3.1.1. Registers.....	82
3.1.2. Reset.....	84
3.1.3. Memory.....	84
3.1.4. Operating Modes.....	87
<b>3.2. The Software Development Process.....</b>	<b>87</b>
<b>3.3. ARM Cortex-M Assembly Language.....</b>	<b>90</b>
3.3.1. Syntax.....	90
3.3.2. Addressing Modes and Operands.....	92
3.3.3. Memory Access Instructions.....	97
3.3.4. Logical Operations.....	99
3.3.5. Shift Operations.....	101
3.3.6. Arithmetic Operations.....	103
3.3.7. Stack.....	120
3.3.8. Functions and Control Flow.....	122
3.3.9. Keil Assembler Directives.....	124
3.3.10. First Example Project.....	125
<b>3.4. Simplified Machine Language Execution.....</b>	<b>129</b>
<b>3.5. CISC versus RISC.....</b>	<b>133</b>
<b>3.6. Details Not Covered in this Book.....</b>	<b>134</b>
<b>3.7. Exercises.....</b>	<b>136</b>
<b>4. Introduction to Input/Output.....</b>	<b>139</b>
4.1. Texas Instruments MSP432 I/O pins.....	140
4.2. Basic Concepts of Input and Output Ports.....	146
4.2.1. I/O Programming and the Direction Register.....	147
4.2.2. Switch Inputs and LED Outputs.....	152
4.3. Clock System.....	162
4.4. SysTick Timer.....	165
4.5. Standard I/O Driver and the printf Function.....	169
4.6. Debugging monitor using an LED.....	173
4.7. Performance Debugging.....	174

4.7.1. Instrumentation .....	174
4.7.2. Measurement of Dynamic Efficiency .....	175
<b>4.8. Exercises .....</b>	<b>177</b>
<b>4.9. Lab Assignments.....</b>	<b>177</b>
<b>5. Modular Programming.....</b>	<b>179</b>
<b>5.1. C Keywords and Punctuation .....</b>	<b>180</b>
<b>5.2. Modular Design using Abstraction .....</b>	<b>184</b>
5.2.1. Definition and Goals .....	184
5.2.2. Functions, Procedures, Methods, and Subroutines.....	186
5.2.3. Dividing a Software Task into Modules .....	187
5.2.4. How to Draw a Call Graph.....	190
5.2.5. How to Draw a Data Flow Graph.....	192
5.2.6. Top-down versus Bottom-up Design .....	193
<b>5.3. Making Decisions .....</b>	<b>194</b>
5.3.1. Conditional Branch Instructions .....	194
5.3.2. Conditional <code>if-then</code> Statements.....	196
5.3.3. <code>switch</code> Statements .....	202
5.3.4. While Loops .....	203
5.3.5. Do-while Loops.....	204
5.3.6. For Loops .....	205
<b>5.4. *Assembly Macros.....</b>	<b>206</b>
<b>5.5. *Recursion.....</b>	<b>208</b>
<b>5.6. Writing Quality Software .....</b>	<b>213</b>
5.6.1. Style Guidelines .....	213
5.6.2. Comments.....	217
5.6.3. Inappropriate I/O and Portability .....	219
<b>5.7. How Assemblers Work.....</b>	<b>219</b>
<b>5.8. Functional debugging .....</b>	<b>221</b>
5.8.1. Stabilization.....	221
5.8.2. Single Stepping .....	221

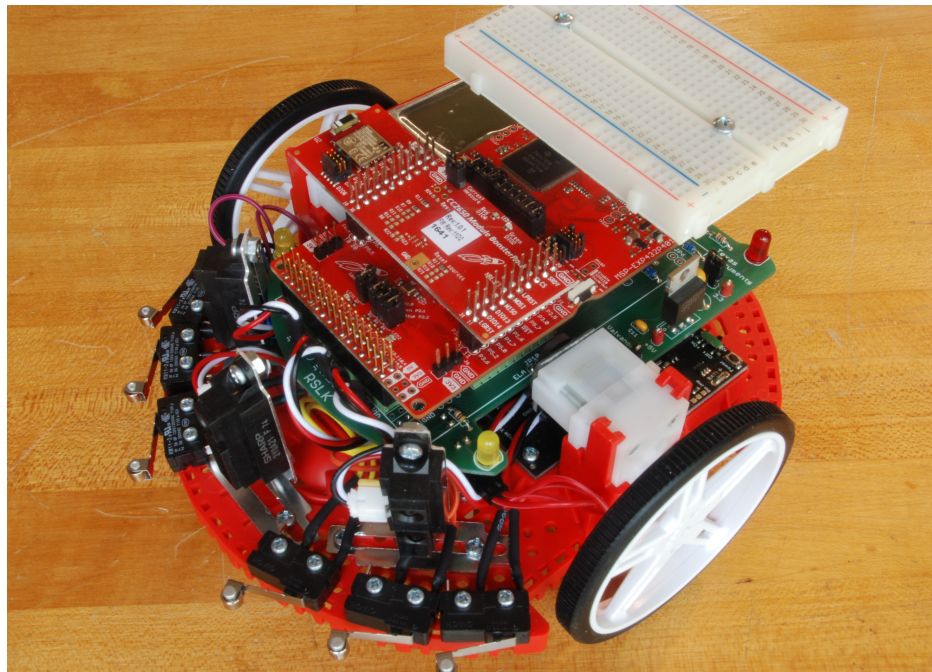
5.8.3. Breakpoints with Filtering.....	222
5.8.4. Instrumentation: Print Statements .....	222
5.8.5. Desk checking .....	223
<b>5.9. Exercises.....</b>	<b>223</b>
<b>5.10. Lab Assignments.....</b>	<b>225</b>
<b>6. Pointers and Data Structures .....</b>	<b>227</b>
<b>6.1. Indexed Addressing and Pointers.....</b>	<b>228</b>
<b>6.2. Arrays .....</b>	<b>231</b>
<b>6.3. Strings .....</b>	<b>236</b>
<b>6.4. Structures .....</b>	<b>238</b>
<b>6.5. Finite State Machines with Linked Structures .....</b>	<b>240</b>
6.5.1. Abstraction .....	240
6.5.2. Moore Finite State Machines.....	241
6.5.3. Mealy Finite State Machines .....	245
<b>6.6. *Dynamically Allocated Data Structures .....</b>	<b>247</b>
6.6.1. *Fixed Block Memory Manager.....	248
6.6.2. *Linked List FIFO .....	249
<b>6.7. Matrices and Graphics .....</b>	<b>252</b>
<b>6.8. *Tables.....</b>	<b>257</b>
<b>6.9. Functional Debugging.....</b>	<b>260</b>
6.9.1. Instrumentation: Dump into Array without Filtering .....	260
6.9.2. Instrumentation: Dump into Array with Filtering. ....	261
<b>6.10. Exercises.....</b>	<b>262</b>
<b>6.11. Lab Assignments.....</b>	<b>264</b>
<b>7. Variables, Numbers, and Parameter Passing.....</b>	<b>265</b>
<b>7.1. Local versus global.....</b>	<b>266</b>
<b>7.2. Stack rules .....</b>	<b>268</b>
<b>7.3. Local variables allocated on the stack .....</b>	<b>270</b>
<b>7.4. Stack frames .....</b>	<b>272</b>
<b>7.5. Parameter Passing .....</b>	<b>275</b>

7.5.1. Parameter Passing in C.....	276
7.5.2. Parameter Passing in Assembly Language.....	278
7.5.3. C Compiler Implementation of Local and Global Variables.....	281
<b>7.6. Fixed-point Numbers .....</b>	<b>283</b>
<b>7.7. Conversions .....</b>	<b>287</b>
<b>7.8. *IEEE Floating-point numbers .....</b>	<b>290</b>
<b>7.9. Exercises .....</b>	<b>295</b>
<b>7.10. Lab Assignments.....</b>	<b>298</b>
<b>8. Serial and Parallel Port Interfacing.....</b>	<b>299</b>
<b>8.1. General Introduction to Interfacing.....</b>	<b>300</b>
<b>8.2. Universal Asynchronous Receiver Transmitter (UART).....</b>	<b>302</b>
8.2.1. Asynchronous Communication .....	304
8.2.2. MSP432 UART Details.....	306
8.2.3. UART Device Driver.....	308
<b>8.3. Synchronous Peripheral Interface, SPI.....</b>	<b>309</b>
<b>8.4. Nokia 5110 Graphics LCD Interface .....</b>	<b>314</b>
<b>8.5. Scanned Keyboards .....</b>	<b>316</b>
<b>8.6. Binary actuators .....</b>	<b>317</b>
8.6.1. Interface .....	317
8.6.2. Electromagnetic and Solid State Relays .....	319
8.6.3. Solenoids.....	320
8.6.4. DC Motor Interface.....	321
<b>8.7. *Pulse-width modulation.....</b>	<b>322</b>
<b>8.8. *Stepper motors.....</b>	<b>323</b>
<b>8.9. Exercises .....</b>	<b>328</b>
<b>8.10. Lab Assignments.....</b>	<b>330</b>
<b>9. Interrupt Programming and Real-time Systems.....</b>	<b>331</b>
9.1. I/O Synchronization .....	331
9.2. Interrupt Concepts.....	336
9.3. Interthread Communication and Synchronization.....	340

<b>9.4. NVIC on the ARM Cortex-M Processor .....</b>	<b>342</b>
<b>9.5. Edge-triggered Interrupts .....</b>	<b>347</b>
<b>9.6. SysTick Periodic Interrupts .....</b>	<b>353</b>
<b>9.7. Periodic Interrupts using TimerA.....</b>	<b>356</b>
<b>9.8. Hardware debugging tools .....</b>	<b>358</b>
<b>9.9. Profiling.....</b>	<b>359</b>
9.9.1 Profiling using a software dump to study execution pattern.....	359
9.9.2. Profiling using an Output Port.....	360
9.9.3. *Thread Profile.....	361
<b>9.10. Exercises.....</b>	<b>363</b>
<b>9.11. Lab Assignments.....</b>	<b>364</b>
<b>10. Analog I/O Interfacing .....</b>	<b>365</b>
<b>10.1. Approximating continuous signals in the digital domain.....</b>	<b>365</b>
<b>10.2. Digital to Analog Conversion .....</b>	<b>367</b>
<b>10.3. Music Generation .....</b>	<b>369</b>
<b>10.4. Analog to Digital Conversion .....</b>	<b>373</b>
10.4.1. MSP432 ADC14 details.....	374
10.4.2. ADC Resolution.....	377
<b>10.5. Real-time data acquisition .....</b>	<b>378</b>
<b>10.6. Exercises.....</b>	<b>379</b>
<b>10.7. Lab Assignments.....</b>	<b>380</b>
<b>11. Communication Systems.....</b>	<b>381</b>
<b>11.1. Introduction.....</b>	<b>381</b>
<b>11.2. Reentrant Programming and Critical Sections .....</b>	<b>384</b>
<b>11.3. Producer-Consumer using a FIFO Queue .....</b>	<b>389</b>
11.3.1. Basic Principles of the FIFO Queue.....	389
11.3.2. FIFO Queue Analysis.....	395
11.3.3. FIFO Queue Implementation.....	396
11.3.4. Double Buffer .....	399
<b>11.4. Serial port interface using interrupt synchronization .....</b>	<b>400</b>



<b>11.5. *Distributed Systems.....</b>	<b>402</b>
<b>11.6. Exercises.....</b>	<b>405</b>
<b>11.7. Lab Assignments.....</b>	<b>407</b>
<b>11.8. Best Practices.....</b>	<b>407</b>
<b>Appendix 1. Glossary.....</b>	<b>408</b>
<b>Appendix 2. Solutions to Checkpoints.....</b>	<b>421</b>
<b>Appendix 3. How to Convert Projects from Keil to CCS.....</b>	<b>429</b>
<b>Appendix 4. Assembly Reference.....</b>	<b>431</b>
<b>Index.....</b>	<b>462</b>



This could be used to accompany Texas Instruments Robot Systems Learning Kit. For more information on RSLK, see <https://training.ti.com/ti-robotics-system-learning-kit>