

Professional's Guide to .NET Micro Framework Application Development

Copyright © 2012 Annabooks, All Rights Reserved

TABLE OF CONTENTS

COPYRIGHT INFORMATION

DEDICATION

FORWARD

ACKNOWLEDGEMENTS

ANNABOOKS / SJJ EMBEDDED MICRO SOLUTIONS

SECTION 1 – INTRODUCTION AND C# BASICS

- 1 WELCOME
 - 1.1 THE EVOLUTION OF PROGRAMMING LANGUAGES
 - 1.2 .NET FRAMEWORK DOWN TO .NET MICRO FRAMEWORK
 - 1.3 WHAT IS AN EMBEDDED SYSTEM?
 - 1.4 WHAT CAN .NET MICRO FRAMEWORK BE USED FOR?
 - 1.5 ABOUT THE BOOK, TARGET AUDIENCE, AND EXAMPLE SOURCE CODE
 - 1.5.1 Section 1 Introduction and C# Basics
 - 1.5.2 Section 2 - .NET Micro Framework SDK and the Process Loop
 - 1.5.3 Section 3 – Hardware I/O
 - 1.5.4 Section 4 – Graphics and WPF
 - 1.5.5 Section 5 - Storage
 - 1.5.6 Section 6 - Projects
 - 1.6 WEBSITE
 - 1.7 SUMMARY: HAVE FUN!
- 2 DEVELOPMENT SYSTEM SETUP AND TARGET HARDWARE CHOICES
 - 2.1 DEVELOPMENT SYSTEM REQUIREMENTS
 - 2.1.1 Basic System Requirements
 - 2.1.2 .NET MF Development Tools
 - 2.1.3 USB Dongle Recommendations
 - 2.2 TARGET PLATFORM GENERAL REQUIREMENTS:
 - 2.3 .NET MF READY PLATFORMS
 - 2.4 EXTRA HARDWARE
 - 2.4.1 Parts List
 - 2.4.2 Suppliers
 - 2.4.3 Other Equipment
 - 2.4.4 Schematic Capture
 - 2.5 SUMMARY: GETTING STARTED
- 3 .NET MICRO FRAMEWORK OVERVIEW
 - 3.1 A BRIEF BACKGROUNDER
 - 3.2 INTRO TO THE CLR

- 3.3 .NET MF ARCHITECTURE
 - 3.3.1 Application Layer
 - 3.3.2 Class Library Layer
 - 3.3.3 CLR
 - 3.3.4 Hardware Layer
- 3.4 OOP PUT INTO PRACTICE
- 3.5 THE POWER OF REUSABLE CODE
- 3.6 C#, JITTING, AND GARBAGE COLLECTION
- 3.7 REAL-TIME MULTITASKING KERNEL VS. CLR
- 3.8 C# VERSUS VB.NET
- 3.9 C# OPERATORS
- 3.10 VALUE TYPES
- 3.11 LOOPS
 - 3.11.1 Do - While
 - 3.11.2 While
 - 3.11.3 For
 - 3.11.4 Foreach, in
- 3.12 PROGRAM FLOW / CONDITIONAL BRANCHING
 - 3.12.1 If-else
 - 3.12.2 Switch-Case
 - 3.12.3 Goto
 - 3.12.4 Return
 - 3.12.5 Try-Catch / Finally
- 3.13 ARRAYS
- 3.14 SUMMARY: PROGRAMMING EVOLVED

SECTION 2 – .NET MICRO FRAMEWORK SDK AND THE PROCESS LOOP

- 4 THE .NET MICRO FRAMEWORK SDK AND THE FIRST .NET MICRO FRAMEWORK APPLICATION
 - 4.1 .NET MF SDK
 - 4.1.1 Install Folders
 - 4.1.2 MFDeploy and Other Tools
 - 4.1.3 References and Namespaces
 - 4.1.4 Sample Projects
 - 4.2 EXERCISE 401: HELLO WORLD
 - 4.2.1 Part 1 – Write the Application
 - 4.2.2 Part 2 – Build and test the application
 - 4.3 DEBUG CLASS AND DOWNLOAD APPLICATIONS
 - 4.4 DEPLOYMENT ISSUES
 - 4.5 SUMMARY: INEXPENSIVE DEVELOPMENT

- 5 INNER PROCESS MECHANICS
 - 5.1 THE INNER PROCESSING MECHANICS - THREADS, PRIORITIES, ROUND-ROBIN, AND INTERRUPTS.
 - 5.1.1 Thread Scheduling
 - 5.1.2 Creating Threads
 - 5.1.3 Priority
 - 5.1.4 Thread Synchronization

- 5.2 EXERCISE 501: LOOKING AT PROCESSING MECHANICS IN ACTION
- 5.3 SUMMARY: ARCHITECT THE IMAGE!

SECTION 3 – HARDWARE I/O

- 6 GENERAL PURPOSE INPUT/OUTPUT PINS
 - 6.1 GPIOS ON DIFFERENT PLATFORMS AND THE HARDWARE LIBRARY
 - 6.2 GPIO CLASSES, ENUMERATORS
 - 6.2.1 GPIO Port Classes
 - 6.2.2 GPIO Pin Enumerations
 - 6.3 EXERCISE 601: TWIDDLE A BIT – FLASH THE ONBOARD LED FOR EDK/EDKPLUS
 - 6.3.1 EX601 Create and Set Up the Project for the EDK/EDKplus
 - 6.3.2 EX601 Adding the Code
 - 6.3.3 EX601 Configure Project Properties; Build, and Deploy the Application.
 - 6.3.4 EX601 Other Hardware Code Listings
 - 6.4 STATIC AND INSTANCE MEMBERS
 - 6.5 EXERCISE 602: BASIC INPUT / OUTPUT – NETDUINO
 - 6.5.1 EX602 Create and Set Up the Project
 - 6.5.2 EX602 Adding the Code
 - 6.5.3 EX602 Configure Project Properties, Build, and Deploy the Application
 - 6.5.4 EX602 Other Hardware Code Listings
 - 6.6 EXERCISE 603: POLLING GPIO PINS ON THE MERIDIAN/P
 - 6.6.1 EX 603 Create and Set Up the Project
 - 6.6.2 EX603 Adding the Code
 - 6.6.3 EX603 Configure Project Properties, Build, and Deploy the Application
 - 6.6.4 EX 603 Other Hardware Code Listings
 - 6.7 EXERCISE 604: GPIO INTERRUPT EXAMPLE USING THE NETDUINO / NETDUINO PLUS
 - 6.7.1 EX604 Create and Set Up the Project
 - 6.7.2 EX604 Adding the Code
 - 6.7.3 EX604 Configure Project Properties, Build, and Deploy the Application.
 - 6.7.4 EX604 Debugging the Interrupt
 - 6.7.5 EX 604 Other Hardware Code Listings
 - 6.8 EXERCISE 605: INTERRUPTS AND THE .NET MF PROCESS LOOP
 - 6.9 EXERCISE 606 – INTERRUPT LATENCY
 - 6.10 GPIO BLOCK ACCESS LIBRARY EXAMPLE FOR THE EDK
 - 6.11 SUMMARY: GPIO OVERLOAD
- 7 SERIAL PORTS
 - 7.1 SERIAL
 - 7.1.1 UART
 - 7.1.2 RS-232
 - 7.1.3 RS-422
 - 7.1.4 RS-485
 - 7.2 SERIALPORT CLASSES
 - 7.3 EXERCISE 701: BASIC RS-232 COMMUNICATION APPLICATION
 - 7.3.1 EX701 - Create the Serial_Test Application
 - 7.3.2 EX701 - Add the Code to Create an Instance of the Serial Port and Echo Characters

- 7.3.3 EX701 - Configure Project Properties, Build, and Deploy the Application
- 7.3.4 EX 701 Other Hardware Code Listings
- 7.4 EXERCISE 702 – GPS TO UART APPLICATION
 - 7.4.1 Parallax™ GPS Receiver Module Summary
 - 7.4.2 Connecting the Module
 - 7.4.3 EX702 Create and Set Up the Project
 - 7.4.4 EX702 Adding the Code
 - 7.4.5 EX702 Configure Project Properties, Build, and Deploy the Application
 - 7.4.6 EX702 FEZ Panda II Solution
 - 7.4.7 EX702 Meridian/P Solution and a Cheaper GPS Alternative
- 7.5 VIRTUAL COM PORTS – SUPPORTING DEVICES THAT ARE NOT PART OF THE .NET MF SDK
- 7.6 SUMMARY: SERIAL PORTS AND MORE SERIAL PORTS

- 8 SERIAL PERIPHERAL INTERFACE (SPI)
 - 8.1 THE SERIAL PERIPHERAL INTERFACE (SPI) BUS, NOT FOR BIT BANG!
 - 8.2 SPI CONNECTIVITY BASICS
 - 8.3 SPI DATA TRANSFER IN ACTION
 - 8.4 THE .NET MF SPI CLASSES
 - 8.5 EXERCISE 801: BASIC SPI LCD PROJECT FOR THE EDK/EDKPLUS
 - 8.5.1 EX801 SPI LCD Controller Commands
 - 8.5.2 Connecting the SPI LCD to the iPac-9302
 - 8.5.3 EX801 Creating a New Application Using the SJJ_MF Console Application Template
 - 8.5.4 EX801 Adding the Code to Send Data to the SPI LCD
 - 8.5.5 EX801 Configure Project Properties, Build, and Deploy the Application.
 - 8.6 EXERCISE 802: SPI LCD SOLUTION USING THE NETDUINO AND MERDIAN/P
 - 8.6.1 EX802 Connecting the SPI LCD to the Netduino or Meridian/P
 - 8.6.2 EX802 Develop and Run the SPI LCD Application for the Netduino
 - 8.6.3 EX802 Develop and Run the SPI LCD Application for the Meridian/P
 - 8.7 EXERCISE 803: MANAGED CODE DRIVER LIBRARY - SERIAL LCD DISPLAY
 - 8.7.1 EX803 Create the SPI_LCD Driver Library
 - 8.7.2 EX803 Adding the Code to the library
 - 8.7.3 EX803 Create a Test Application for iPac 9302
 - 8.7.4 EX803 Create a Test Application for Netduino
 - 8.7.5 EX803 Create a Test Application for Meridian/P
 - 8.8 EXERCISE 804: SPI READ - SPI KEYPAD INPUT FOR EDK
 - 8.8.1 EX804 Hardware Setup
 - 8.8.2 Ex804 Creating the SPI Keypad Application for the EDK
 - 8.8.3 EX804 Build, Deploy, and Test
 - 8.9 MORE SERIAL PORTS! SPI-TO-SERIAL 16 BIT TRANSFER EXAMPLE
 - 8.9.1 Overview
 - 8.9.2 The Hardware
 - 8.9.3 The Application
 - 8.9.4 Managed Code Driver
 - 8.9.5 Continuous Transfer Mode Workaround
 - 8.10 THE SPI SUMMARY

- 9 PULSE WIDTH MODULATION (PWM)

- 9.1 PULSE-WIDTH MODULATION OVERVIEW
- 9.2 PWM PROGRAMMATIC ACCESS
 - 9.2.1 The V4.2 .NET MF PWM Class Properties
 - 9.2.2 Vendor Custom PWM Support
- 9.3 EXERCISE 901: PWM TEST FOR THE MERIDIAN/P
 - 9.3.1 EX 901 Create and Set Up the project
 - 9.3.2 EX901 Adding the Code
 - 9.3.3 EX901 Configure Project Properties, Build, and Deploy the Application.
- 9.4 EXERCISE 902 – PWM TEST FOR THE NETDUINO
 - 9.4.1 EX902 Create and Set Up the Project
 - 9.4.2 EX902 Adding the Code
 - 9.4.3 EX902 Configure Project Properties, Build, and Deploy the Application.
- 9.5 EDK/EDKPLUS PWM MANAGED CODE DRIVER API
 - 9.5.1 Internal SOC PWM Controller
 - 9.5.2 CPLD PWM Controllers
 - 9.5.3 Virtual COM Port
- 9.6 EXERCISE 903 PWM TEST FOR THE EDK / EDKPLUS
 - 9.6.1 EX903 Create and Set Up the Project
 - 9.6.2 EX903 Add the Code
 - 9.6.3 EX903 Configure Project Properties, Build, and Deploy the Application
- 9.7 EXERCISE 904: A SECOND PWM TEST FOR THE EDK
 - 9.7.1 EX904 Create and Set Up the Project
 - 9.7.2 EX904 Add the Code to Create the PWM Driver Object and Green LED GPIO Object
 - 9.7.3 EX904 Configure Project Properties, Build, and Deploy the Application
 - 9.7.4 EX904 Running the Application
- 9.8 SUMMARY: PULSE WIDTH MODULATION (PWM)

- 10 ANALOG-TO-DIGITAL CONVERTER (ADC)
 - 10.1 ANALOG-TO-DIGITAL CONVERTER OVERVIEW
 - 10.2 ADC PROGRAMMATIC ACCESS
 - 10.2.1 The V4.2 .NET MF AnalogInput Class Properties
 - 10.2.2 Vendor Custom ADC Support
 - 10.3 EDK/EDKPLUS ADC MANAGED CODE DRIVER API
 - 10.3.1 EDK/EDKplus' ADC Channels
 - 10.3.2 ADC Managed Code Driver API
 - 10.4 EXERCISE 1001: BASIC ADC TEST FOR EDK/EDKPLUS
 - 10.4.1 EX1001 Create the ADC_Test Application
 - 10.4.2 EX 1001 Add the Code to Create the ADC Driver Object and the Green LED GPIO Object.
 - 10.4.3 EX1001 Configure Project Properties, Build, and Deploy the Application
 - 10.4.4 EX1001 Running ADC_Test
 - 10.5 EXERCISE 1002 BASIC TEST FOR NETDUINO / NETDUINO PLUS
 - 10.5.1 EX1002 Create and Set Up the Project
 - 10.5.2 EX1002 Adding the Code
 - 10.5.3 EX1002 Configure Project Properties, Build, and Deploy the Application.
 - 10.5.4 EX1002 FEZ Panda II Code Listing
 - 10.6 SUMMARY: ANALOG-TO-DIGITAL CONVERTER (ADC)

- 11 ETHERNET
 - 11.1 ETHERNET HISTORY
 - 11.2 SOCKETS
 - 11.2.1 Socket Class
 - 11.2.2 Socket Setup
 - 11.3 EXERCISE 1101 REMOTE CONTROL – NETDUINO PLUS
 - 11.3.1 EX1101 Create and Set Up the Project
 - 11.3.2 EX1101 Adding the Code
 - 11.3.3 EX1101 Configure Project Properties, Build, and Deploy the Application.
 - 11.4 EXERCISE 1102: ADD HTTP SERVER SUPPORT – NETDUINO PLUS
 - 11.4.1 EX1102 Modify the EX1101 Project
 - 11.4.2 EX1102 Configure Project Properties, Build, and Deploy the Application
 - 11.5 EXERCISE 1103 SERVER ACCESS – LOOPBACK – EDK/EDKPLUS
 - 11.5.1 The Desktop Client Application
 - 11.5.2 Socket Server Application
 - 11.5.3 Create the Socket Server Application
 - 11.5.4 Adding the Socket Server Code
 - 11.5.5 Configure Project Properties, Build, and Deploy the Application
 - 11.5.6 Running the Application
 - 11.6 SUMMARY: ETHERNET, TCP/IP, AND SOCKETS

SECTION 4 – GRAPHIC USER INTERFACE

- 12 BITMAP GUI
 - 12.1 BITMAP CLASS
 - 12.2 EXERCISE 1201 – SCREEN TEST
 - 12.3 EXERCISE 1202 – DRAW A BITMAP
 - 12.4 EXERCISE 1203 - FONT CONVERSION
 - 12.4.1 Example 1: Convert One of the SDK Fonts
 - 12.4.2 Example 2: Convert Different 3rd Party Font
 - 12.5 EXERCISE 1204 - ADD TEXT TO BITMAP
 - 12.6 EXERCISE 1205 - BITMAP ANIMATION
 - 12.7 SUMMARY: BASIC GRAPHICS

- 13 WINDOWS PRESENTATION FOUNDATION CLASS
 - 13.1 APPLICATION CLASS
 - 13.2 WPF CLASSES
 - 13.2.1 Presentation Namespace Classes
 - 13.2.2 Presentation Controls Namespace Classes
 - 13.2.3 Presentation Media Namespace Classes
 - 13.2.4 Presentation Shapes Namespace Classes
 - 13.3 EXERCISE 1301 – OUT OF BOX EXAMPLE
 - 13.4 EXERCISE 1302 STACK PANEL
 - 13.5 EXERCISE 1303 CANVAS
 - 13.6 EXERCISE 1304 LIST BOX MULTI-WINDOW EXAMPLE
 - 13.7 SUMMARY: WPF A LITTLE MORE GOING ON

- 14 TOUCHSCREEN AND INK
 - 14.1 TOUCHSCREEN AND INK CLASSES
 - 14.2 SDK TOUCHSCREEN AND INK EXAMPLES
 - 14.3 EXERCISE 1401 – ADDING TOUCH SUPPORT TO THE LIST BOX MULTI-WINDOW EXAMPLE
 - 14.4 FROM GUI TO HARDWARE AND BACK
 - 14.5 EX1402 GUI AND HARDWARE INTERACTION
 - 14.6 SUMMARY: USER INTERACTION / USER FRIENDLY

SECTION 5 – STORAGE

- 15 FLASH STORAGE
 - 15.1 FILE STORAGE SUPPORT
 - 15.2 FILE CLASS
 - 15.3 STREAMING CLASSES
 - 15.4 VOLUMEINFO CLASS
 - 15.5 EXERCISE 1501 – FILE CLASS BASIC CREATE AND SAVE A FILE
 - 15.5.1 EX1501 Create and Set Up the Project
 - 15.5.2 EX1501 Adding the Code
 - 15.5.3 EX1501 Configure Project Properties, Build, and Deploy the Application
 - 15.5.4 EX1501 FEZ Panda II Code Listing
 - 15.6 EXERCISE 1502 – FILE CLASS WRITE AND READ
 - 15.6.1 EX1502 Update EX1501
 - 15.6.2 EX1502 Configure Project Properties, Build, and Deploy the Application
 - 15.7 EXERCISE 1503 – STREAMWRITER EXAMPLE
 - 15.7.1 EX1503 Create and Set Up the Project
 - 15.7.2 EX1503 Adding the Code
 - 15.7.3 EX1503 Configure Project Properties, Build, and Deploy the Application
 - 15.7.4 EX1503 FEZ Panda II Code Listing
 - 15.8 SUMMARY: DATA STORAGE
- 16 EXTENDED WEAK REFERENCES
 - 16.1 MEMORY MANAGEMENT (GARBAGE COLLECTION) AND WEAK REFERENCES
 - 16.2 DATA STORAGE WITH EXTENDED WEAK REFERENCES
 - 16.3 EWR CLASS
 - 16.4 EXERCISE 1601: A SIMPLE EWR EXAMPLE
 - 16.4.1 Create the Storage Application
 - 16.4.2 Build, Deploy, and Test
 - 16.5 SUMMARY: NICE IDEA FOR SCRATCH PAD INFORMATION

SECTION 6 – PROJECTS

- 17 MIGRATE APPLICATIONS TO A NEW PLATFORM
 - 17.1 CODE ONCE AND RUN EVERYWHERE, BUT...
 - 17.2 EXERCISE 1701 – KEYPAD APPLICATIONS MIGRATION TO MERIDIAN/P
 - 17.3 NEW VISUAL STUDIO AND .NET MF SDK MIGRATION
 - 17.4 SUMMARY: APPLICATION LIFE CYCLE
- 18 ROBOT PROJECT: TRAXSTER II

- 18.1 THE GENESIS OF THE .NET MF AND TRAXSTER
 - 18.2 THE TRAXSTER ROBOT BUILD
 - 18.3 WHY THE SERIALIZER?
 - 18.4 THE SERIALIZER COMMAND LIBRARY
 - 18.5 THE MAIN TRAXSTER II PROGRAM
 - 18.6 SUMMARY: SOMETHING TO BUILD-ON
- 19 ROBOT PROJECT 2: STINGER
- 19.1 PROJECT GOAL: AUTONOMOUS GPS NAVIGATION ROVER
 - 19.2 LAYING OUT THE ROBOT SUBSYSTEMS AND POWER REQUIREMENTS
 - 19.3 THE STINGER ROBOT BUILD
 - 19.3.1 Robot Body
 - 19.3.2 Push-Button LED Piezo Connections
 - 19.3.3 Serializer Connection
 - 19.3.4 GPS Circuit
 - 19.3.5 Camera Connection
 - 19.3.6 4x20 SPI-LCD (Optional)
 - 19.3.7 Other Device Connections
 - 19.3.8 Final Power Connection
 - 19.4 SOFTWARE SUBSYSTEMS TESTS
 - 19.4.1 Buttons / LEDs Test
 - 19.4.2 GPS Driver and Test
 - 19.4.3 4x20 SPI-LCD Driver and Test
 - 19.4.4 Ambient Temperature Test
 - 19.4.5 Servo Test
 - 19.4.6 Basic Stinger Test
 - 19.5 FULL SYSTEM TEST: PUTTING IT ALL TOGETHER
 - 19.6 SUMMARY: ALMOST THERE
- A BIBLIOGRAPHY
- A.1 ARTICLES AND WHITE PAPERS
 - A.2 BOOKS:
 - A.3 CONFERENCES
 - A.4 TRAINING COURSES
 - A.5 WEBSITES
 - A.6 HELP FILES

ABOUT THE AUTHORS