Table of Contents

Preface 9
Why you should buy this book 9
What is parametric programming? 10
A word about CAM systems 10
Scope 10
Versions of Custom Macro 10
Machine types 10
Prerequisites 11
Optional status 11
Lessons structure 11

Lesson 1 - Introduction to Custom Macro 13
Objectives 13
Introduction 13
Comparison to sub-programming 13
Comparison to canned cycles 14
Comparison to computer programming 15
All programs can use Custom Macro functions 15
Application categories 15
Part-families 15
How are your drawings dimensioned? 16
Part-family program structure 17
User-created canned cycles 18
Utilities 20
An automatic tool changer test 20
A part counter 20
A jaw boring program 20
A manual corner pick-up utility 21
Complex motions and geometric shapes 21
Need more motion types? 22
Driving accessory devices 23
Limitations 23
Commonly an option that must be purchased 24
Compatibility 24
Custom Macro programs must reside in the control 24
Execution time for certain commands 24
Program preparation time 24
Skill required 24
Introduction to feature types 25
Computer-related features (lessons 2-5) 25
Variables (lesson 2) 25
Arithmetic (lesson 3) 25
Logic (lesson 4) 25
Looping (lesson 5) 25
CNC-related features (lessons 6-7) 25
Access to machine position (lesson 6) 25
Access to offsets (lesson 6) 25
Access to current modes (lesson 6) 25
Alarm generation (lesson 6) 26
Creation of G-, M-, and T-codes (lesson 7) 26
Access to control panel functions (lesson 6) 26
Timers (lesson 6) 26
Access to parameters (lesson 2)) 26
Surprised? 26
Key Points for Lesson 1: 26
Quiz 26

Lesson 2 - Variables 29
Objectives 29
Introduction 29
What are variables? 29
Comparison to offsets 29
Flexibility of variables 30
Comparison to memories in a calculator 30
Variables types 31
Arguments 31
Arguments with user created canned cycle applications 31
Arguments with part-family applications 34
Local variables 36
Local variables with argument assignment number one 36
Local variables with argument assignment number two 38
How do you specify which argument assignment type you are using? 39
More on vacancy 39
Keeping track of local variables 40
Local variables at different program nest levels 40
Secondary purpose for local variables: general purpose calculations 41
Displaying local variable values 42
Common variables 42
To minimize redundant calculations 42
For general purpose calculations 42
As counters 43
As flags 43
Displaying common variables 43
Permanent common variables 43
Do you have a touch-probing system? 44
Creating your own system constants 44
Displaying permanent common variables 45
Offsets as variables 45
Write-protecting permanent common variables 45
Labeling permanent common variables 45
System variables 46
Summary of Custom Macro variables 46
Key Points for Lesson 2: 46
Quiz 46

Lesson 3 - Arithmetic 49
Objectives 49
Introduction 49

Basic functions 49
Combining operations to form an expression 49
Referencing the value of a previously assigned variable 50
Referencing a variable's value within its own assigning command 50
Advanced arithmetic functions 51
Trigonometry functions 51
An application 52
A note about older versions of Custom Macro 53
Rounding functions 53
Applications for rounding 54
Other arithmetic functions 55
Square root 55
Absolute value 55
Power 56
Add decimal point 56
Natural logarithm and exponent using base e (2.718…) 58
Binary and binary coded decimal conversions 58
Reading parameter values 58
What if you do not have a needed arithmetic function? 59
Priority of arithmetic operations 59
Summary of arithmetic capabilities 60
Basic functions: 60
Advanced functions 60
Example programs 61
About program organization 61
Milling an outside rectangular shape 61
A one-plunge grooving cycle 64
An internal thread milling cycle 66
Warning about the example programs 68
Key Points for Lesson 2: 68
Quiz 69

Lesson 4 - Logic and Program Flow Control 71
Objectives 71
Introduction 71
Statement labels 71
Unconditional branching 72
Unconditional branching versus program execution time 72
Applications for unconditional branching 72
Changing the machining order 73
Using optional block skip to exit a program 75
With conditional branching 76
Conditional branching 76
Syntax for conditional branching statements 76
The conditional expression 76
The universal conditional branching command format 76
Using THEN for true conditions 78
Testing with multiple logical expressions using AND and OR 79
Applications for conditional branching 80
Argument flags 80
To control coolant 80
To climb or conventional mill 81
More possibilities for argument flags 81
Global flags 81
Argument error trapping 82
Testing mandatory arguments 82
Testing for erroneous argument settings 84
What about program execution time? 84
Setting default values 84
Other kinds of error trapping 85
Looping 86
Conditional branching example 86
Key Points for Lesson 4: 89
Quiz 89

Lesson 5 - Creating Loops 91
Objectives 91
Introduction 91
Steps to looping 91
The IF statement loop explained 93
The WHILE statement loop explained 93
Which is better? 93
Getting stuck in a loop 93
Creating multiple loops in a Custom Macro 94
Nesting Loops 95
Solving challenges related to loops 96
When is a loop required? 96
To minimize commands 96
When the number of loop repetitions is itself a variable 97
How many times must the loop be executed? 97
What else must be initialized and stepped? 99
Are there any other constants required for the loop? 100
Are approach motions required prior to the loop? 100
What must be done during each loop execution? 100
Loop examples 102
Bolt-hole circle example 102
Multiple-pass grooving example 104
Nesting loops 107
A grid pattern example 107
A taper thread milling Custom Macro program 110
Key Points for Lesson 5: 112
Quiz 113

Lesson 6 - System Variables 115
Objectives 115
Introduction 115

Introducing system variables 116
- Variations in system variable numbering 116

Access to tool offsets 116
- Machining center tool length and cutter radius compensation offsets 116
  - The advantage of using the system variable name (instead of its number) 119
- Machining center workpiece coordinate system offsets 119
- Turning center wear offsets 120
- Turning center geometry offsets 121
- A trick with system variable numbers 121
- Do you have more offsets? 123
- Applications for offset access 123
  - As extra variables 123
  - Improving upon compensation types 123
  - Adding wear offsets to machining centers that do not have them 124
  - Automating sizing adjustments during unattended operation 125
  - Transferring wear offsets to geometry offsets after first workpiece inspection 126
  - Streamline the testing of machining center offsets 127
  - Simplifying sizing adjustments 128
  - User-created compensation types 129
  - Other ideas? 131

Access to current position 132
- Understanding with and without compensation 132
- About axis number 133
- For a three axis machining center 133
- For a two axis turning center: 134
- Applications for axis position access 134
  - Assuming initial position in user-created canned cycles 134
  - A tool length measuring Custom Macro program (machining centers) 136
  - A program zero measuring and assigning Custom Macro program (machining centers) 137

Access to alarm generation 138
- A part counting Custom Macro program 139
- A tool life manager with a difference 139
- A pallet changer testing program 142
- Minimizing the time required for error trapping 143
  - Using optional block skip 145
  - Using a check sum test 146

Stop with message 148
- Applications for stop with message 149

Suppressing single block and waiting for auxiliary functions 149
- An application for suppressing single block 150
- An application for not waiting for M-codes 150

Suppressing feed hold, feed-rate override, and exact stop check 151
- All-in-one system variable 151
- Separate system variable for each function 151
- Applications 151
  - For feed hold and feed-rate override 151

Access to timer and clock (date/time) 152
- Timing events with the timers 152
  - The beginning of a time-based tool life monitor 152
- Understanding the values for date and time 153

Access to input and output signals 154

Access to modal information 155
- Modal G-code access for machining centers 155
  - An application 156
- Modal words for machining centers 156
- For turning centers 157

Other system variables of note 157
- A test to confirm system variable numbers and names 157
- Null system variable 157
- Other system constants 157
  - PI (3.14159265358979323846) 157
  - Base of natural logarithm (2.71828182845904523536O 157
- Work shift offsets on turning centers 157
- Access to commonly changed parameters 158
- Status of mirror image 158
- Status during restart of a program 158
- Total number of parts and number of parts required 158
- Type of tool compensation memory 158
- Main program number 158
- Access to currently instated offset values 158
- A few more system variables 159

Key Points for Lesson 6: 159

Quiz 159

Lesson 7 - Other Features of Custom Macro 161

Objectives 161

Introduction 161

Protecting important programs 161

Naming permanent common variables 162
- Notes about SETVN: 162

Modal calling commands 163
- Modal calling command G66.1 163
- Modal calling command G66 166

Creating user-defined G-, M-, and T-codes 167
- Creating user-defined G-codes 167
  - Creating a series of user-defined G-codes 169
- Creating user-defined M-codes 169
  - M-codes allowing letter-address arguments 169
  - M-codes not allowing arguments 170
  - Creating a series of user-defined M-codes 171

Creating user-defined T-codes 171
Creating user-defined sub-program calls with other letter-addresses 172
Applications for user-defined G-, M-, and T-codes 173
Making programs for similar machines more compatible 173
Commanding two or more functions from one M-code 174
Improving the automatic tool changer command for machining centers 174
Error trapping an automatic tool changer obstruction 174
Interfacing an accessory device with input/output system variables (#1000 and #1100 series) 175
Changing the function of commands with which you do not agree 175
Creating G-codes to make a FANUC control compatible to another control 177
Monitoring the tool in the spindle (user-defined T-code application) 178
Totaling spindle run time 180
Sending data out through the serial communications port 180
Backing up workpiece coordinate system offsets 181
The application 181
Conclusion to other CNC-related features of Custom Macro 183
Understanding argument assignment number two 183
Local variable assignments used with argument assignment number two 183
How does the control know the difference? 184
When to use argument assignment number two 184
A circular cam application 185
How will the Custom Macro know when to stop? 186
Other potential applications for argument assignment number two 187
Key Points for Lesson 7: 188
Quiz 188

Lesson 8 - Approaching and Verifying Custom Macro Applications 189
Objectives 189
Introduction 189
Approaching applications 190
Approaching part-family applications 190
Step one - Develop a process that can be used for the entire family 191
Step two - Choose the required arguments 192
Step three - Be sure you have solved all problems related to the application 197
Step four - Write the Custom Macro program 197
Step five - Error trap potential mistakes 200
Step six - Eliminate the execution of redundant commands 202
Approaching user-created canned cycle applications 205
Step one - Determine what your user-created canned cycle must do 206
Step two - Consider default values and flags 208
Step three - Consider error trapping 208
Step four - Be sure you have solved all of the problems 208
Step five - Write the Custom Macro program 209
Approaching complex motions and shapes applications 210
Step one - Solve the arithmetic problems of your application 212
Step two - Determine your arguments 213
Step three - Write an example calling program 214
Step four - Write the Custom Macro program 214
Approaching utility applications 215
Locate a challenge and come up with a solution 215
Approaching accessory device applications 217
A spindle touch-probe 217
Verifying Custom Macro programs 218
Mistakes beginners are prone to making 218
Syntax mistakes 218
Simple evaluation mistakes 221
A drastic mistake in your solution to the problem 221
Key Points for Lesson 8: 221

Appendix A - Two Detailed Part-Family Application Examples 223
Objectives 223
Introduction 223
Defining ideal part-family applications 223
Why are you handling your part-family application with Custom Macro? 224
Process and workpiece revisions 224
Consistency 224
Separating process from workpiece definition 225
Getting the setup person involved with programming 225
Our example part-family (machining center example) 225
How will arguments be specified? 227

More on input data 228
Use lots of system constants 229
Keep track of variables 230
Error trap input variables 230
Saving program execution time 230
Another suggestion for input variables 233
Local versus common variable usage 235
What if you're running out of variables? 236
Handling raw material issues 238
Handling cutting tool issues 241
Handling work holding and program zero assignment issues 244
Getting ready to create the machining commands 246
Tool startup program 246
Tool ending program 248
Machining commands 248
Program organization 251
Protecting programs 252
Managing programs 252
Going from job to job 252
Documentation issues: 252
A template part definition program 253
Cutting tool documentation 254
Cutting conditions documentation 254
Work-holding setup and program zero documentation 254
Drawings and instructions 254
Turning center example 255
Machining (main) program 255
The part definition program 256
The get everything ready program 257
The system constants program 258
The error trapping program 258
Cutting conditions program 259
Tool station selection program 261
Clear all common variables program 261
Tool startup program 262
Tool ending program 262
Machining programs 262
Machining center example using the second type of part definition program 266
Handling what-ifs 268
What if the four slots are too deep too completely machine in on pass? 268
What if some of the workpieces in the family don't have slots? 269
What if some left hand tools are used? 269
What if some materials used do not allow coolant? 272
What if some of the raw material comes in the form of solid bar, not tubing? 273
Verifying your part-family application 275
Using machine flags 275
More about error trapping 276
Incorporating current Custom Macro programs in your part-family application 276
Program listings 283
Machining center example using first method of part definition: 283
Turning center example: 287

Appendix B - Introduction to Spindle Touch-Probes 301
Objectives 301
Introduction 301
Skip cutting command (G31) 301
Understanding overshoot and droop 301
Skip-signal position system variables 302
The corner pick-up program revisited 302
Is it good enough? 303
Applications 303
A trial machining spindle touch-probing program 304

Programming Activities 305