Text Encoding Simplifies Microcontroller Command Parsing

DAVID HUNTER | FIRST CONSULTING INC. drhunter@frontiernet.net

WHILE WORKING ON a Microchip PIC project, I created a set of SCPI-style (Standard Commands for Programmable Instruments) commands to control the PIC. These SCPI commands use the first four characters of text words separated by a colon.

In previous projects, I found that parsing text consumes significant computing time and code space. Typically, text parsing is handled by string comparisons or developing a parsing tree. Neither of these techniques is simple to design and implement on a microcontroller.

I knew that it would be faster to parse commands if I could convert the text into 16-bit numbers. So, I developed a method that converts the first four characters of each command to upper case and then encodes them as a 16-bit number. Each character is translated into a 4-bit representation and then packed into a 16-bit number.

But don't you need 5 bits to represent 26 letters? Yes, if each letter is treated uniquely. To reduce the letters to 4 bits, I analyzed two-letter pairs and grouped the letters based on how often they are used. This encoding worked out well for the 25 or so commands I needed. (More extensive command sets may need to be checked for duplication and the encoding changed accordingly.)

ENCODING CHART

4-bit code	Character	4-bit code	Character
0000	<space>	1000	L, T
0001	A	1001	N, R
0010	E	1010	B, D
0011	I	1011	C, G
0100	O	1100	K, P
0101	U	1101	F, H
0110	Y	1110	M, V, W
0111	S	1111	J, Q, X, Z

The encoding gives <space>, A, E, I, O, U, Y, and S single codes since they are very common. The consonants are then grouped together in sets. The table shows the encoding for the letters. One implementation in C with the space character handled separately is:

```
const unsigned char LookUpTable[] = {0x1,0xA,0xB,0xA,0x2,
0xD,0xB,0xD, 0x3,0xF,0xC,0x8,0xE,0x9,0x4,0xC,0xF,0x9,0x7,
0x8,0x5,0xE,0xE,0xF,0x6,0xF};
```

These two examples show the encoding of SCPI commands:

CLS<space> translates to 0xB870
CALCulate:AVERage:COUNt translates to
0xB18B,0x1E29,0xB459

DAVID HUNTER is an electrical engineer with First Consulting Inc. in Rochester, N.Y. He has a BSEE and an MSEE from the Rochester Institute of Technology and has worked for more than 25 years as a design engineer in embedded-systems software, digital, analog, and RF circuit hardware design.

After encoding the incoming text, parsing is just a matter of checking 16-bit numbers rather than text strings. This can be done as a CASE statement or series of IF statements, either of which is much simpler (and usually faster) than handling text strings in a microcontroller. Using this approach greatly reduced the amount of code needed.

IDEAS FOR DESIGN WANTED

Send us your Ideas For Design. We'll pay you $150 for every Idea For Design that we publish. In addition, this year's top design as selected by our readers will earn an additional $500, with two runners-up each receiving $250. You can submit your Ideas For Design via:

- E-mail: richard.gawel@penton.com

OR BY

- Postal mail to:
  Ideas For Design
  Electronic Design
  1166 Avenue of the Americas, 10th Floor
  New York, NY 10036

Go to *www.electronicdesign.com* for our submission guidelines.