Knowledge Exchange

Using Look Up Tables

by Sean King, September 2008

Abstract

Knowing how to use look up tables is a valuable skill for any embedded programmer, they allow us to replace potentially complex runtime computation with a simple array indexing operation. This article offers ways to apply these look up tables in Flowcode and offers examples demonstrating these techniques.

Requirements

Software:

- Professional or Site licence of Flowcode v3 or v4 for any variant.

Hardware:

- *EB043 Graphical LCD* (for examples)

In some applications look-up tables can be used as an alternative to complex calculations. If the results of a calculation are known, for all the possible values of the input variables, they can be stored in the memory of a microcontroller (memory limits allowing) and accessed under the control of the input variable(s).
The data can be obtained from other programs (spreadsheets, data loggers, mathematical CAD packages, etc.) and imported into the Flowcode program as data arrays.

As a simple example with a single input variable, some applications might require the values of a sine wave at 1 degree intervals. Calculation of these values during the execution of a program will be time consuming, and require large amounts of program and data memory. The values can be pre-calculated using a spreadsheet and imported into an array, in the correct order, allowing the angle value to be used as the array index.
The attached Excel program shows an example of the generation of the C arrays required for PIC, AVR and ARM programs, using sine wave data. The floating-point values are multiplied by 100 and rounded so they can be stored and manipulated as single bytes. The appropriate array code can be copied and pasted directly into a C code block in a Flowcode program (changing the array name if required).

Most microcontrollers allow data to be stored in three separate locations.

- Data memory (RAM)
- Non-volatile data memory (EEPROM)
- Program memory (FLASH)

Until recently, data stored in program memory could not be modified after downloading. Many newer devices do now allow program memory to be modified by the program, though then process is usually slow and complicated.
Data stored in the data EEPROM is slow to write, but quick and easy to read, and will not be lost when the power is removed.
Data stored in RAM is quick and easy to read and write, but is volatile, and must be re-calculated each time power is applied to the device. This may still be useful if the calculations can be performed during an initialisation period.

With the sine wave example we only need to store one quadrant of values (0 to 90 degrees). Data can be accessed in a way that will allow the generation of all the sine, cosine and tangent values, using only simple calculations. Other examples might operate adequately using interpolation to calculate values between the ones stored in the array.

Program commands

Flowcode directly supports look-up tables in both data memory (as strings/arrays) and EEPROM (using the EEPROM component). The program memory option requires the use of C code blocks.

The format of the command to store up to 256 bytes of data in the program memory of a PIC device is:

PIC: **rom char* table_name = {d0, d1,d2,d3,d4,.........,d255};**

The format of the commands to store data in the program memory of an ARM or AVR device is:

ARM: **const char table_name[] = {d0, d1,d2,d3,d4,.........,dxxx};**

AVR: **const char table_name[] PROGMEM = {d0, d1,d2,d3,d4,.........,dxxx};**

The AVR and ARM array sizes are only limited by the available memory.

Accessing an individual element of the data table requires the following code:

FCV_DATA_VAL = table_name [FCV_INDEX]; (PIC, ARM)
FCV_DATA_VAL = pgm_read_byte(table_name + FCV_INDEX); (AVR)

The AVR code also requires you to include the code *#include <avr\pgmspace.h>* in the supplementary code option in the Project Options in Flowcode. Where FCV_DATA_VAL and FCV_INDEX are the C code versions of the variable names 'data_val' and 'index', declared as byte variables in Flowcode.
Note: The same variables can be accessed either from C or Flowcode using the appropriate version of the variable name.

Examples:

As part of this article there are three example Flowcode files along with an excel spreadsheet that shows how to create look up tables using Flowcode. The Excel file is used to create the data for the Look up table. Once the data has been created it can be included into Flowcode by using the Edit -> Supplementary code dialogue window.

Example 1 - SIN Look Up Table

Uses a look up table to convert a degree value between 0 and 180 into a value of SIN with amplitude of 100.

Example 2 - SIN look up table using the EB048 Gfx LCD.

As above but now the program steps through the Lookup table to output a Sine wave onto the Graphics LCD. Note that there are visible steps between the LUT values rather then a smooth continuous line.

Example 3 - SIN look up table using using the EB048 Gfx LCD with Interpolation.

As above but now the steps have disappeared as the output is a combination of the Look up table and interpolation between the LUT values.

Further reading

Below are some links to other resources and articles on related subjects, and technical documentation relating to the hardware used for this project...

Flowcode:	http://www.matrixmultimedia.com/flowcode.php
E-blocks	http://www.matrixmultimedia.com/eblocks.php
Learning Centre:	http://www.matrixmultimedia.com/lc_index.php
User Forums:	http://www.matrixmultimedia.com/mmforums
Product Support:	http://www.matrixmultimedia.com/sup_menu.php