

```

// StockFun.cpp - STOCK PRICE DISPLAY FUNCTIONS
//
// USAGE
// StockDisp text-file
//
// text-file      is the name of a file containing the definition of the
//                message to be displayed.
//
// MAINTENANCE HISTORY
// DATE          PROGRAMMER AND DETAILS
// 07-10-04      JS          Original
//
//-----

#include <sys/types.h>          // System type declarations
#include <ctype.h>              // Character type functions
#include <time.h>               // System time declarations
#include <string.h>             // String manipulation functions
#include <signal.h>             // Signal codes
#include <unistd.h>             // Operating system standard functions
#include <fcntl.h>              // File control options
#include <errno.h>              // Operating system error codes
#include <time.h>               // Time declarations
#include <sys/time.h>           // Interval timer declarations
#include <sys/socket.h>         // Socket declarations
#include <netinet/in.h>         // Internet type conversion functions
#include <arpa/inet.h>          // More Internet type conversion functions
#include <sys/ipc.h>            // IPC declarations
#include <sys/shm.h>            // Shared memory declarations
#include <sys/sem.h>            // Semaphore declarations
#include <iostream>             // C++ input/output streams
#include <iomanip>               // C++ input/output manipulators
#include <fstream>              // File stream declarations
#include <sstream>              // String stream declarations
#include "StockDef.h"           // Stock Price Display declarations
using namespace std;           // Expand standard namespace to global scope

//-----

// FONT RECORD STRUCTURE

struct FontRec_t {
    unsigned short    fontColCnt;          // Column count
    short             fontColDataOffset;  // Offset of column data
};

//-----

static const unsigned short COL_DATA[] = {
    /* 20 */ 0x0000,0x0000,0x0000,
    /* 21 */ 0x1FE8,
    /* 22 */ 0x1E00,0x0000,0x1E00,
    /* 23 */ 0x0040,0x0278,0x03C0,0x1E40,0x0278,0x03C0,0x1E40,0x0200,
    /* 24 */ 0x0710,0x0888,0x1FFC,0x0888,0x0670,
    /* 25 */ 0x0C00,0x1210,0x1220,0x0C40,0x0080,0x0100,0x0230,0x0448,0x0848,0x0030,
    /* 26 */ 0x0070,0x0E88,0x1108,0x1188,0x0E68,0x0010,0x0028,
    /* 27 */ 0x1E00,
    /* 28 */ 0x03F8,0x0C06,0x1001,
    /* 29 */ 0x1001,0x0C06,0x03F8,
    /* 2A */ 0x0A00,0x0600,0x1C00,0x0600,0x0A00,
    /* 2B */ 0x0080,0x0080,0x03E0,0x0080,0x0080,
    /* 2C */ 0x0004,0x0008,
    /* 2D */ 0x0080,0x0080,0x0080,0x0080,
    /* 2E */ 0x0008,
    /* 2F */ 0x000E,0x0070,0x0380,0x1C00,
    /* 30 */ 0x0FF0,0x1008,0x1008,0x1008,0x0FF0,
    /* 31 */ 0x0800,0x0800,0x1FF8,
    /* 32 */ 0x0C38,0x1048,0x1088,0x1108,0x0E08,
    /* 33 */ 0x0810,0x1008,0x1108,0x1108,0x0EF0,
    /* 34 */ 0x00C0,0x0140,0x0640,0x0840,0x1FF8,0x0040,
    /* 35 */ 0x1F90,0x1108,0x1108,0x1108,0x10F0,
    /* 36 */ 0x0FF0,0x1088,0x1108,0x1108,0x08F0,
    /* 37 */ 0x1000,0x1038,0x10C0,0x1300,0x1C00,
    /* 38 */ 0x0EF0,0x1108,0x1108,0x1108,0x0EF0,
    /* 39 */ 0x0F10,0x1088,0x1088,0x1108,0x0FF0,
    /* 3A */ 0x0208,
    /* 3B */ 0x0004,0x0208,
    /* 3C */ 0x0080,0x01C0,0x0360,0x0630,0x0410,
    /* 3D */ 0x0140,0x0140,0x0140,0x0140,0x0140,
    /* 3E */ 0x0410,0x0630,0x0360,0x01C0,0x0080,
    /* 3F */ 0x0C00,0x1000,0x10E8,0x1100,0x0E00,
    /* 40 */

```

```

0x03C0,0x0C30,0x1008,0x1008,0x21C4,0x23E4,0x23E4,0x13E0,0x11A0,0x0C20,0x03C0,
/* 41 */ 0x0018,0x00E0,0x0740,0x1840,0x0740,0x00E0,0x0018,
/* 42 */ 0x1FF8,0x1108,0x1108,0x1108,0x1108,0x0EF0,
/* 43 */ 0x07E0,0x0810,0x1008,0x1008,0x1008,0x0810,
/* 44 */ 0x1FF8,0x1008,0x1008,0x1008,0x0FF0,
/* 45 */ 0x1FF8,0x1108,0x1108,0x1108,0x1008,
/* 46 */ 0x1FF8,0x1100,0x1100,0x1100,0x1000,
/* 47 */ 0x0FF0,0x1008,0x1008,0x1088,0x1088,0x08F0,
/* 48 */ 0x1FF8,0x0100,0x0100,0x0100,0x0100,0x1FF8,
/* 49 */ 0x1FF8,
/* 4A */ 0x0030,0x0008,0x0008,0x1FF0,
/* 4B */ 0x1FF8,0x0100,0x0280,0x0440,0x0820,0x1010,0x0008,
/* 4C */ 0x1FF8,0x0008,0x0008,0x0008,0x0008,
/* 4D */ 0x1FF8,0x0600,0x0180,0x0060,0x0018,0x0060,0x0180,0x0600,0x1FF8,
/* 4E */ 0x1FF8,0x0800,0x0600,0x0100,0x00C0,0x0030,0x1FF8,
/* 4F */ 0x07E0,0x0810,0x1008,0x1008,0x0810,0x07E0,
/* 50 */ 0x1FF8,0x1080,0x1080,0x1080,0x0F00,
/* 51 */ 0x07E0,0x0810,0x1008,0x1028,0x0810,0x07E8,
/* 52 */ 0x1FF8,0x1080,0x1080,0x10C0,0x0F38,
/* 53 */ 0x0E30,0x1108,0x1108,0x1088,0x0C70,
/* 54 */ 0x1000,0x1000,0x1000,0x1FF8,0x1000,0x1000,0x1000,
/* 55 */ 0x1FE0,0x0010,0x0008,0x0008,0x0010,0x1FE0,
/* 56 */ 0x1800,0x0700,0x00E0,0x0018,0x00E0,0x0700,0x1800,
/* 57 */
0x1E00,0x01E0,0x0018,0x00E0,0x0700,0x1800,0x0700,0x00E0,0x0018,0x01E0,0x1E00,
/* 58 */ 0x1818,0x0660,0x0180,0x0180,0x0180,0x0660,0x1818,
/* 59 */ 0x1800,0x0400,0x0300,0x00F8,0x0300,0x0400,0x1800,
/* 5A */ 0x1018,0x10E8,0x1708,0x1808,
/* 5B */ 0x1FFC,0x1004,
/* 5C */ 0x1C00,0x0380,0x0070,0x000E,
/* 5D */ 0x1004,0x1FFC,
/* 5E */ 0x0800,0x1000,0x2000,0x1000,0x0800,
/* 5F */ 0x0002,0x0002,0x0002,0x0002,0x0002,0x0002,0x0002,
/* 60 */ 0x1000,0x0800,
/* 61 */ 0x0170,0x0288,0x0288,0x0288,0x01F0,0x0008,
/* 62 */ 0x1FF8,0x0110,0x0208,0x0208,0x01F0,
/* 63 */ 0x01F0,0x0208,0x0208,0x0208,0x0110,
/* 64 */ 0x01F0,0x0208,0x0208,0x0208,0x0110,0x1FF8,
/* 65 */ 0x01F0,0x0248,0x0248,0x0248,0x01D0,
/* 66 */ 0x0200,0x0FF8,0x1200,
/* 67 */ 0x01F0,0x0209,0x0209,0x0111,0x03FE,
/* 68 */ 0x1FF8,0x0100,0x0200,0x0200,0x01F8,
/* 69 */ 0x13F8,
/* 6A */ 0x0001,0x13FE,
/* 6B */ 0x1FF8,0x0040,0x00C0,0x0120,0x0210,0x0008,
/* 6C */ 0x1FF8,
/* 6D */ 0x03F8,0x0100,0x0200,0x0200,0x01F8,0x0100,0x0200,0x0200,0x01F8,
/* 6E */ 0x03F8,0x0100,0x0200,0x0200,0x01F8,
/* 6F */ 0x01F0,0x0208,0x0208,0x0208,0x01F0,
/* 70 */ 0x03FF,0x0110,0x0208,0x0208,0x0208,0x01F0,
/* 71 */ 0x01F0,0x0208,0x0208,0x0208,0x0110,0x03FF,
/* 72 */ 0x03F8,0x0100,0x0200,
/* 73 */ 0x0190,0x0248,0x0248,0x0248,0x0130,
/* 74 */ 0x0200,0x0FF0,0x0208,
/* 75 */ 0x03F0,0x0008,0x0008,0x0010,0x03F8,
/* 76 */ 0x0380,0x0070,0x0008,0x0070,0x0380,
/* 77 */ 0x0300,0x00E0,0x0018,0x00E0,0x0300,0x00E0,0x0018,0x00E0,0x0300,
/* 78 */ 0x0208,0x0110,0x00E0,0x00E0,0x0110,0x0208,
/* 79 */ 0x0301,0x00C1,0x0032,0x000C,0x0030,0x00C0,0x0300,
/* 7A */ 0x0218,0x0268,0x0288,0x0308,
/* 7B */ 0x0040,0x0040,0x0FBE,0x1001,0x1001,
/* 7C */ 0x1FFF,
/* 7D */ 0x1001,0x1001,0x0FBE,0x0040,0x0040,
/* 7E */
0x0380,0x0C60,0x3098,0x41C4,0x40E4,0x8072,0x8062,0x80C2,0x4184,0x4104,0x3218,0x0C60,0
x0380,
};

```

```

static const FontRec_t FONT_REC_ARR[] = {
    { 3, 0}, /* */
    { 1, 3}, /* ! */
    { 3, 4}, /* " */
    { 8, 7}, /* # */
    { 5, 15}, /* $ */
    { 10, 20}, /* % */
    { 7, 30}, /* & */
    { 1, 37}, /* ' */
    { 3, 38}, /* ( */
    { 3, 41}, /* ) */
    { 5, 44}, /* * */
    { 5, 49}, /* + */
    { 2, 54}, /* , */

```

```

{ 4, 56}, /* - */
{ 1, 60}, /* . */
{ 4, 61}, /* / */
{ 5, 65}, /* 0 */
{ 3, 70}, /* 1 */
{ 5, 73}, /* 2 */
{ 5, 78}, /* 3 */
{ 6, 83}, /* 4 */
{ 5, 89}, /* 5 */
{ 5, 94}, /* 6 */
{ 5, 99}, /* 7 */
{ 5, 104}, /* 8 */
{ 5, 109}, /* 9 */
{ 1, 114}, /* : */
{ 2, 115}, /* ; */
{ 5, 117}, /* < */
{ 5, 122}, /* = */
{ 5, 127}, /* > */
{ 5, 132}, /* ? */
{ 11, 137}, /* @ */
{ 7, 148}, /* A */
{ 5, 155}, /* B */
{ 6, 160}, /* C */
{ 5, 166}, /* D */
{ 5, 171}, /* E */
{ 5, 176}, /* F */
{ 6, 181}, /* G */
{ 6, 187}, /* H */
{ 1, 193}, /* I */
{ 4, 194}, /* J */
{ 7, 198}, /* K */
{ 5, 205}, /* L */
{ 9, 210}, /* M */
{ 7, 219}, /* N */
{ 6, 226}, /* O */
{ 5, 232}, /* P */
{ 6, 237}, /* Q */
{ 5, 243}, /* R */
{ 5, 248}, /* S */
{ 7, 253}, /* T */
{ 6, 260}, /* U */
{ 7, 266}, /* V */
{ 11, 273}, /* W */
{ 6, 284}, /* X */
{ 7, 290}, /* Y */
{ 4, 297}, /* Z */
{ 2, 301}, /* [ */
{ 4, 303}, /* \ */
{ 2, 307}, /* ] */
{ 5, 309}, /* ^ */
{ 7, 314}, /* _ */
{ 2, 321}, /* ` */
{ 6, 323}, /* a */
{ 5, 329}, /* b */
{ 5, 334}, /* c */
{ 6, 339}, /* d */
{ 5, 345}, /* e */
{ 3, 350}, /* f */
{ 5, 353}, /* g */
{ 5, 358}, /* h */
{ 1, 363}, /* i */
{ 2, 364}, /* j */
{ 6, 366}, /* k */
{ 1, 372}, /* l */
{ 9, 373}, /* m */
{ 5, 382}, /* n */
{ 5, 387}, /* o */
{ 6, 392}, /* p */
{ 6, 398}, /* q */
{ 3, 404}, /* r */
{ 5, 407}, /* s */
{ 3, 412}, /* t */
{ 5, 415}, /* u */
{ 5, 420}, /* v */
{ 9, 425}, /* w */
{ 6, 434}, /* x */
{ 7, 440}, /* y */
{ 4, 447}, /* z */
{ 5, 451}, /* { */
{ 1, 456}, /* | */
{ 5, 457}, /* } */
{ 13, 462}, /* ~ */

```

```

};
const size_t FONT_REC_CNT = sizeof(FONT_REC_ARR) / sizeof(FONT_REC_ARR[0]);
const size_t FONT_REC_OFS = ' ';

//-----
// GLOBAL DATA

Bitmap_t          fontBitmap;    // Font bitmap structure
unsigned short    fontColArr[50]; // Column array

//-----

// SHIFT AND LOAD

void
ShiftAndLoad (
    unsigned short    column)
{
    size_t            i;          // General purpose index

    for (i = 0; i < 16; i++) {
        if ((column & 1) != 0)
            cout << '0';
        else
            cout << '.';
        column >>= 1;
    }
    cout << '\n' << flush;
}

//-----

// GET BITMAP

const Bitmap_t *
GetBitmap (
    char            ch)          // Character to be displayed
{
    int             fontInd;      // Font index
    const FontRec_t *fontRec;    // Font record
    size_t          i;           // General purpose index

    fontInd = ch - FONT_REC_OFS;
    if (fontInd < 0 || fontInd > static_cast<int>(FONT_REC_CNT))
        fontInd = ' ' - FONT_REC_OFS;
    fontRec = FONT_REC_ARR + fontInd;
    for (i = 0; i < fontRec->fontColCnt; i++)
        fontColArr[i] = COL_DATA[fontRec->fontColDataOffset+i];
    fontColArr[i++] = 0;
    fontBitmap.bmColCnt = i;
    fontBitmap.bmColArr = fontColArr;
    return &fontBitmap;
}

```