

# Code Review for Lab 2

Stephen A. Edwards

Columbia University

Fall 2011



```

//Nicole Lewis, Feifei Kong, Vanshil Shah
char keyboard_key()
{
/*Set up the keyboard as a double array of single characters*/
char keyb[7][6] =      {{'N' , 'I' , 'P' , 'M' , 'F' , 'A'},
                        {'C' , 'R' , 'V' , 'B' , '%' , 'L'},
                        {'U' , '(' , ')' , '_' , '<' },
                        {'^' , '7' , '8' , '9' , '/' },
                        {'v' , '4' , '5' , '6' , '*' },
                        {'S' , '1' , '2' , '3' , '-' },
                        { 0 , '0' , '.' , '=' , '+' }};

/*Scan through the entire keyboard to find which key is pressed.*/
int j,i,a,b; //initialize
for(j = 0; j<7; j++) //This loop checks each column
{
keyboard_column_low(j); //Sets the j column to low to check the row.
for (i = 0 ; i < 6 ; i++) //This loop checks for the row that is pressed.
    { if (!keyboard_row_read(i)){ //if the row is low (button is
        // pushed) it sets a and b as the row and column that
        // need to be saved to indicate the button
        a = i;
        b = j;}
    }
keyboard_column_high(j); //Sets the j column back to high so it is
                        // not activated when checking the next column
}
return keyb[b][a]; //returns the character in the double array
}

```

//Nick Duckwiler, Dimitri Dyatlov, Kevin Roark

```
int keyboard_key()
{
    int i,j,n,*a;
    for(j=0;j<COLUMNS;++j){
        for(i=0;i<ROWS;++i){
            for (n = 0; n < COLUMNS; n++)
                { keyboard_column_high(n);
                }
            keyboard_column_low(j);
            if(!keyboard_row_read(i)){
                return i+10*j;
            }
        }
    }
    return -1;
}
```

```

void integer_to_character(int key_value, char *display){
char values[100][10]={ "N", "I/YR", "PV", "PMT", "FV", "Amort", "CshFl",
    "IRR", "NPV", "Bond", "%", "RCL", "INPUT", "(,)", "+/-", "BACKSPACE",
    "UP ARROW", "7", "8", "9", "DIVIDE", "DOWN ARROW", "4", "5", "6",
    "MULTIPLY", "SPACE", "1", "2", "3", "-", "ON/CE", "0", ".", "=", "+"};
int i, j;
if(key_value<29)
    j=(key_value/10)*6+(key_value%10);
else
    j=17+(key_value/10-3)*5+(key_value%10);
for(i=0; i<10; ++i){
*display=values[j][i];
display++;
}
};

int main()
{
    *AT91C_WDTC_WDMR = AT91C_WDTC_WDDIS;
    int i;
    lcd_init();
    keyboard_init();
    char display[10];
    int last_key;

```

```
for(;;){
    while(!(last_key+1)){
        last_key=keyboard_key();
        lcd_print7("no key          ");
    }
    lcd_print7("          ");
    while(last_key+1){
        last_key=keyboard_key();
        integer_to_character(last_key,&display);
        lcd_print7(display);
    }
}

return 0;
}
```

//Ankita Gore, Shikhar Kumar, Christina Huang

```
int keyboard_key()
{
    int r;
    int c;

    for (c = 0; c < 7; c++) { //cols
        keyboard_column_low(c);

        for (r = 0 ; r < 6 ; r++) { //rows

            if (!keyboard_row_read(r)) {
                keyboard_column_high(c);
                return (c*10)+r;      }
        }
        keyboard_column_high(c);
    }
    return -1;
}
```

```
int main()
{
    lcd_init();
    keyboard_init();

    for (;;) {
        int x = keyboard_key();

        if (x==-1) {
            lcd_print7("No key");
        }

        else {
            lcd_print7("KEY ");
            lcd_put_char7('0'+(x/10), 4);
            lcd_put_char7('0'+(x%10), 5);
        }
    }
    return 0;
}
```

```
//Gabriela Melchior - gdm2118
//Alysia Sanchez ams2377
//Jessica Wang jlw2197
//Dong Yeop Kang dk2645
```

```
#define num_rows 6
#define num_cols 7
```

```
int keyboard_key()
{
```

```
    int k; // column
    int i; // row
    const char keys [num_cols][num_rows] =
        {{'N', 'I', 'P', 'M', 'F', 'A'},
         {'C', 'R', 'V', 'B', '%', 'L'},
         {'\r', '(', ')', '~', '\b', 0},
         {'\v', '7', '8', '9', '/', 0},
         {'\n', '4', '5', '6', '*', 0},
         {'S', '1', '2', '3', '-', 0},
         {'0', '0', '.', '=', '+', 0}};
```



```
for (k=0; k<num_cols; k++)//tests columns
{
    keyboard_column_low(k);

    for(i=0; i<num_rows; i++)//tests rows
    {
        if(!keyboard_row_read(i))//if key is pressed
        {
            keyboard_column_high(k);
            return keys[k][i];//return pressed key
        }
    }
    keyboard_column_high(k);//reset
}

return -1;
}
```

```

//Abhinav Mishra Andrew Pope Yiming Ge Anna Teng
// Will VanArsdall Kaiven Zhou
#define COLUMNS 7
#define ROWS 6
const char* keys[44] = {"", "N", "I/YR", "PV", "PMT", "FG", "Amort",
    "CshFl", "IRR", "NPV", "Bond", "%", "RCL",
    "INPUT", "(", ")", "+/-", "<-", "",
    "UP", "7", "8", "9", "/", "",
    "DOWN", "4", "5", "6", "x", "",
    "SHIFT", "1", "2", "3", "-", "",
    "", "0", ".", "=", "+"};

int main()
{
    lcd_init();
    *AT91C_WDTC_WDMR = AT91C_WDTC_WDDIS;
    keyboard_init();

    for (;;) {
        if(keyboard_key()!=0)
            lcd_print7(keys[keyboard_key()]);
        else
            lcd_print7(" ");
    }
    return 0;
}

```

```
int keyboard_key()
{
    int i, j;
    int x = 1;

    for(i = 0; i < COLUMNS; i++)
    {
        keyboard_column_low(i);

        for(j = 0; j < ROWS; j++)
        {
            if(!keyboard_row_read(j))
            {
                keyboard_column_high(i); //Resets the c
                return x;
            }
            x++;
        }

        keyboard_column_high(i);
    }
    return 0;
}
```

```
// Binna Han, Eric Leung, Michael Yan
```

```
int main()
{
    lcd_init();
    keyboard_init();
    for (;;)
    {
        keyboard_init();
        char display = keyboard_key(); // returned char from keyboard_key()
        int display2 = keyboard_key(); // returned int from keyboard_key()
        if (display2 != -1)
        {
            // display when some key is pressed
            lcd_print7("SEE");
            lcd_put_char7(' ', 3);
            lcd_put_char7(display, 4);
            lcd_put_char7(' ', 5);
        }
        else
        {
            lcd_print7("NO KEY");
        }
    }
    return 0;
}
```

```
#define columns 7
#define rows 6
// create array of all keys
const char keysPressed[columns][rows] =
{
    {'N', 'I', 'P', 'M', 'F', 'A'},
    {'C', 'R', 'V', 'B', '%', 'L'},
    {'\r', '(', ')', '~', '\b', 0},
    {'\v', '7', '8', '9', '/', 0},
    {'\n', '4', '5', '6', '*', 0},
    {'S', '1', '2', '3', '-', 0},
    {'0', '0', '.', '=', '+', 0}
};
```

```
int keyboard_key()
{
    // initlaize keyboard, reset every column to high
    keyboard_init();
    int x; // goes through rows
    int i; // goes through columns
    for (x = 0 ; x < columns; x++)
    {
        keyboard_column_low(x);

        for (i = 0 ; i < rows ; i++)
        {
            if (!keyboard_row_read(i))
            {
                return keysPressed[x][i];
            }
        }
    }
    return -1; // no keys are pressed
}
```

```
// Stephen Edwards
#define NUM_COLUMNS 7
#define NUM_ROWS 6
const char keyboard_keys[NUM_COLUMNS][NUM_ROWS] = {
    {'N', 'I', 'P', 'M', 'F', 'A'},
    {'C', 'R', 'V', 'B', '%', 'L'},
    {'\r', '(', ')', '~', '\b', 0},
    {'\v', '7', '8', '9', '/', 0},
    {'\n', '4', '5', '6', '*', 0},
    {'S', '1', '2', '3', '-', 0},
    {0, '0', '.', '=', '+', 0}};
int keyboard_key()
{
    int row, col;
    for (col = 0 ; col < NUM_COLUMNS ; col++) {
        keyboard_column_low(col);
        for (row = 0 ; row < NUM_ROWS ; row++)
            if (!keyboard_row_read(row)) {
                keyboard_column_high(col);
                return keyboard_keys[col][row];
            }
        keyboard_column_high(col);
    }
    return -1;
}
```