Book Corrections

Though I tried to produce perfect books, I found editing your own work can be very difficult. I learned to improve the process with help from outsiders but even that was hard since it's hard to find a good editor that also understands the topic of programming. Then I have to deal with changes to the technology I used in the book such as the compiler and any updates that occur over time. The programming tools can also get updated or outdated. I have no control over this stuff so this corrections summary, or as the industry calls it; the errata sheet, is my attempt to help my readers get through all my typos and mistakes and all the changes that occur beyond my control. As always, I'm here to help via email at chuck@elproducts.com.

Compiler Errors

In the first Beginner's Guide to Embedded C Programming book I used the latest HI-TECH C compiler available at the time which was version 9.60. It loaded with MPLAB 8.02. Volume 2 and Volume 3 used the latest at the time of their writing. The code all maintained compatibility so there weren't many issues.
The Microchip bought out the company that made the HI-TECH compiler and began to make improvements. In version 9.81 those improvements affected my books. Version 9.81 and all later versions include changes to the header files that affect some of the register names and configuration settings. This has caused all the programs in my books to give errors if run as written in the book.

I covered it in detail in my April Newsletter (which you can sign up for free at my website) but here is that section reproduced for you to help you work around this issue.

HI-TECH Version 9.81+

Microchip recently updated the HI-TECH C compiler for 10F, 12F and 16 parts to version 9.81 and in this update they made a change that affects all the code in my Beginner's Guide to Embedded C books. They made a change to the register names to match the data sheet exactly. These definitions are contained in the various header files for the different microchip PICs. This is a good move overall but for those using programs written for the previous versions 9.80 and earlier; this causes many errors to appear when the project is compiled. This has resulted in many emails back to me regarding this exact issue as my books were written to use 9.80 or earlier. There is a simple fix thankfully. By adding the one line at the very top of the program (it has to be the first line) all the original header files are used instead of the new definition files. If you have experienced this issue, this is the fix.

```
#define _LEGACY_HEADERS
```

For example here is how it’s used in a program from my book Beginner's Guide to Embedded C.

```
#define _LEGACY_HEADERS          // Define required for compiler version 9.81
and later
#include <htc.h>                 // Include HITECH CC header file

//Internal clock, Watchdog off, MCLR off, Code Unprotected
__CONFIG (INTIO & WDTDIS & PWRTDIS & MCLRDIS & BORDIS &
UNPROTECT & IESODIS & FCMDIS );

main()
{
ANSEL =     0;                   // Intialize A/D ports off
CM1CON0 = 0;                     // Initialize Comparator 1 off
CM2CON0 = 0;                     // Initialize Comparator 2 off
PORTC = 0x00;                    //Clear PortC port
TRISC = 0x00;                    //All PortC I/O outputs
while(1==1)                      //loop forever
{
        RC0 = 1;                 // Turn on RC0 LED
}         //End while
}         //end main
```

If you don't like this method, there are other options to fix this as well using MPLAB setup screens spelled out in the readme file included with the HI-TECH installation. Some have chosen alternative methods. One person re-wrote the header files back to the original format. It's been interesting how different this was approached by my readers. It also shows there isn't one right way to solve a software issue.

This should address a majority of the issues I get email on but there are a few other errata to spell out for the three Beginner’s Guide to Embedded C Programming books. They are listed below.

Beginner’s Guide to Embedded C Programming (Volume 1)

Correction #1

All programs require the PORTs to be set to digital with these lines:

```
ANSEL =     0;                  // Intialize A/D ports off
CM1CON0 = 0;                    // Initialize Comparator 1 off
CM2CON0 = 0;                    // Initialize Comparator 2 off
```

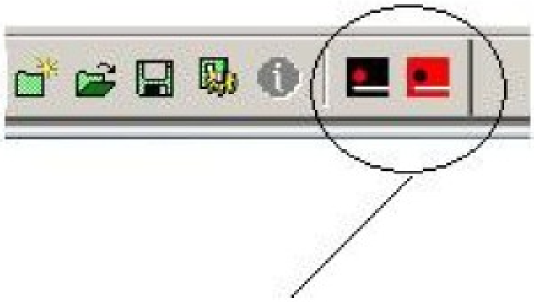
The first couple programs in the book I left this off and the compiler didn’t complain but later versions of the compiler expected these to be included. Add those lines to the top of the program and it should work fine.

These settings are explained in the book on pages 115-117.
I've corrected this on all the files in the .zip file you download at
www.elproducts.com/cbookfiles
This is really the proper way to start a program so I should have done this anyway.

If you cannot get the first project to light the LED then add these lines and it should correct the problem.

Correction #2

When I started the books, the HI-TECH compiler only supported a handful of parts and only limited use of the program memory. Then they released a lite version that supported all the PIC Microcontrollers. It was called PICC PRO Lite. When you install the new PICC PRO Lite version, you will see they changed the look of the icons in MPLAB as shown below. These work the same as the ones shown in the book but just look different.



New PICC PRO ICONs

Correction #3

Page 40: If-Else example

```
If (GP2 == 0)
{
GP1 = 1;
}
else
{
GP2 = 0; <<<<----- Should be GP1 = 0;
}
```

Correction #4

Page 95: **How It Works** section

States:

Notice how it starts with a **backslash** and a star /*.

It should state:

Notice how it starts with a **forward slash** and a star /*.

Correction #5

Page 97: Half way down the page

States:

The comment line that describes the settings follows a double **backslash**.

It should state:

The comment line that describes the settings follows a double **forward slash**.

Correction #6
Page 117 and page 126: Code Example

States:

```
PORTC = 0x00;          // Clear PortB port
TRISC = 0x00;          //All PortB I/O outputs
```

Should state:

```
PORTC = 0x00;          // Clear PortC port
TRISC = 0x00;          //All PortC I/O outputs
```

Correction #7
Page 123: Project 4 description.

The description says the project will scroll light back and forth but the actual code only scrolls in one direction and then repeats it. The LED **light doesn't scroll back and forth** it just scrolls in one direction when you get the project working on your PICkit 2 development board.

Correction #8
Page 131: Typo

States:

You **cold** also change the main loop's

Should state:

You **could** also change the main loop's

Correction #9
Page 16, 2nd paragraph, 6th line

States:
program could look **line** one continuous line

Should state:

program could look **like** one continuous line

Correction 11:
Page 114 and 127
This really isn't an error as 0xD is treated the same as 0x0D in the compiler but it's best to be consistent in a beginner book.

States:

TMR0 = **0xD**

Should state:

TMR0 = **0x0D**

Correction 12:
Page 133, 136 and 137
I described the operation backwards.

States:
When switch is pressed the LED will light.

Should state:

When the switch is idle the LED will light and will be off when the switch is pressed.

Correction #13:
Page 109, 1st paragraph, last sentence

States:
Functions were described earlier in Chapter 6 and this project I use to introduces how to actually use functions.

Should State:
Functions were described earlier in Chapter 6 and **in** this project I **introduce** how to actually use functions.

Correction #14:
Page 113 and 118
Comment states RB0 when should state the LED name on the board.

States:
// Turn on RB0 LED
// Turn off RB1 LED

Should state:
// Turn on DS1 LED
// Turn on DS2 LED

Correction #15:
Page 88, Project 1 description is wrong.

States:

DS1 LED should flash in Project 1

Should state:

DS1 LED stays on all the time in Project 1

Correction #16:
Page 98, Bottom of the page

States:

```
while(1=1)
```

Should state:

```
while(1==1)
```

Beginner’s Guide to Embedded C Programming - Volume 2

Correction #1

Page 16, shows wrong part number for two pin header
Shows this:

Parts List

Qty	Description	Part Number/Supplier
1	Small Breadboard	923273-ND / Digikey
4	Shorting Blocks	22023 / Jameco
5	Two Pin 0.1” Single Row Header	**222966** / Jameco
2	16 Pin 0.1” Female Header (cut to 10 Pin)	308592 / Jameco

Should be this:

Parts List

Qty	Description	Part Number/Supplier
1	Small Breadboard	923273-ND / Digikey
4	Shorting Blocks	22023 / Jameco
5	Two Pin 0.1” Single Row Header	**108338** / Jameco
2	16 Pin 0.1” Female Header (cut to 10 Pin)	308592 / Jameco

Correction #2

Page 31, Comment is incorrect

States:

default: //if state_**led > 3**

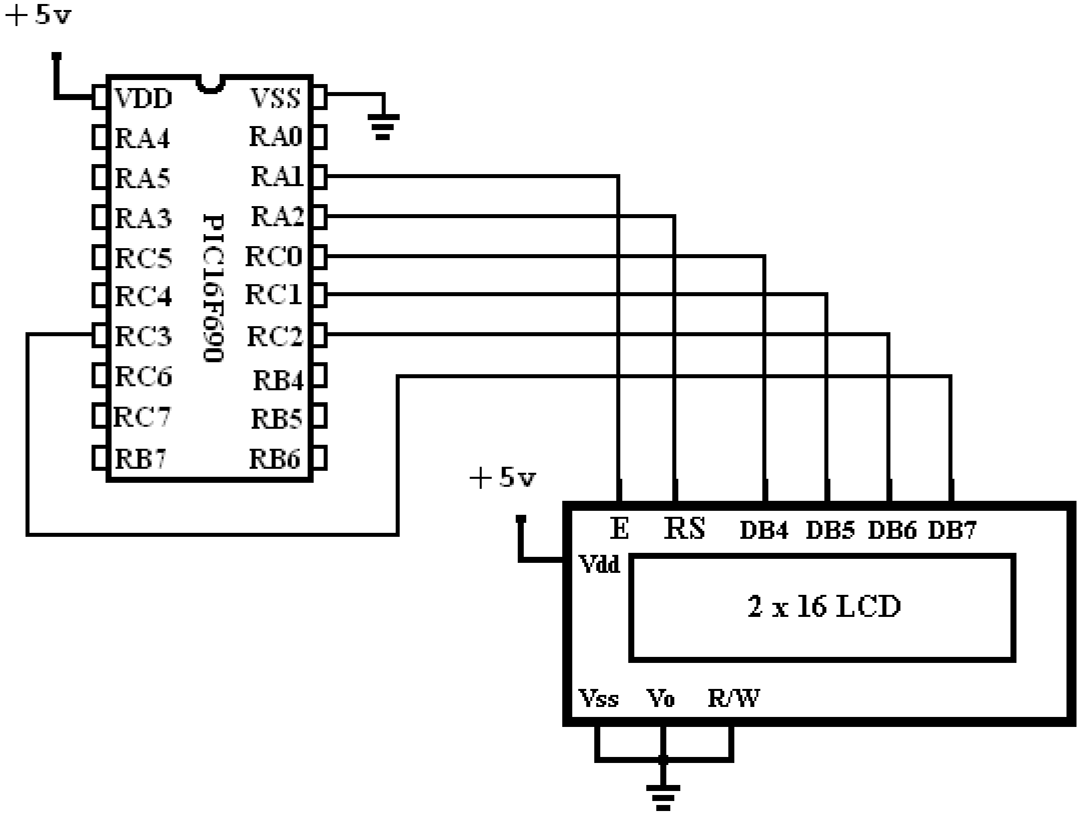
Should be:

default: //if state_**led >9**

Correction #3

Chapter 4, Figure 4-2

RA2 should be connected to LCD RS pin per the schematic below:



Corrected Figure 4-2

Correction #4
Chapter 8, Page 127 and 137

Page 127
Code states: CPK = 1;
Should be: CPK = 0;

Page 137
States:
The clock polarity bit is initially set to idle high but by setting this bit to a one then the clock will idle low.

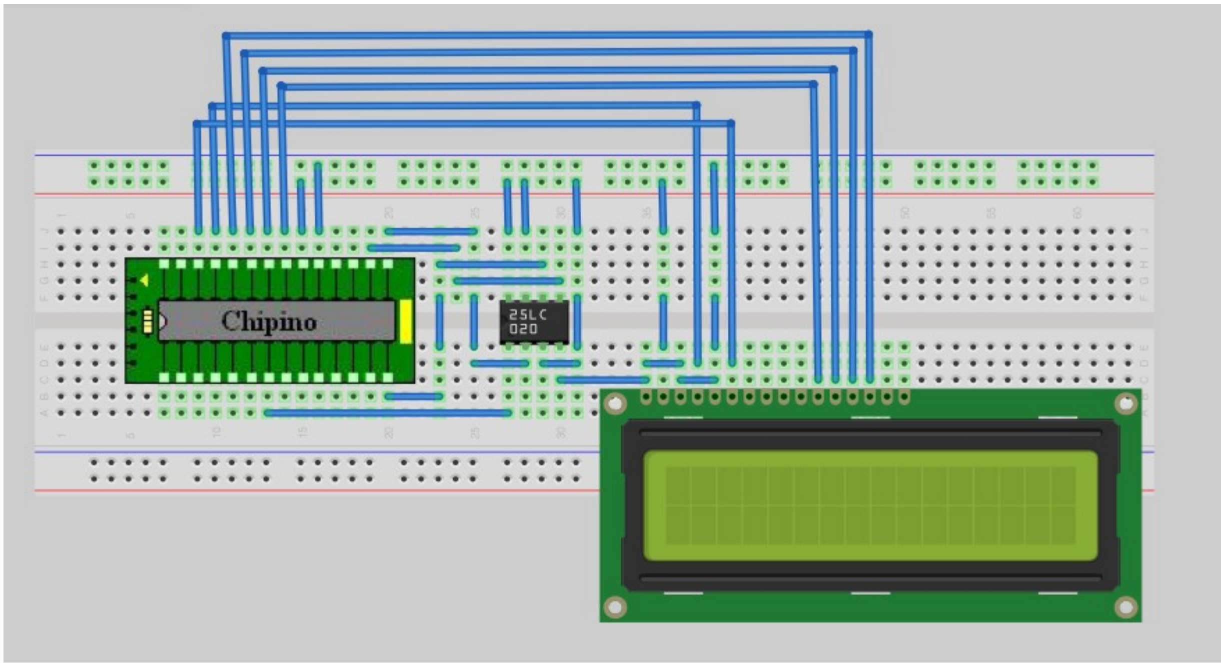
Should be:
The clock polarity bit is initially set to idle high but by setting this bit to a **zero** then the clock will idle low.

Also States:
CKP = 1;

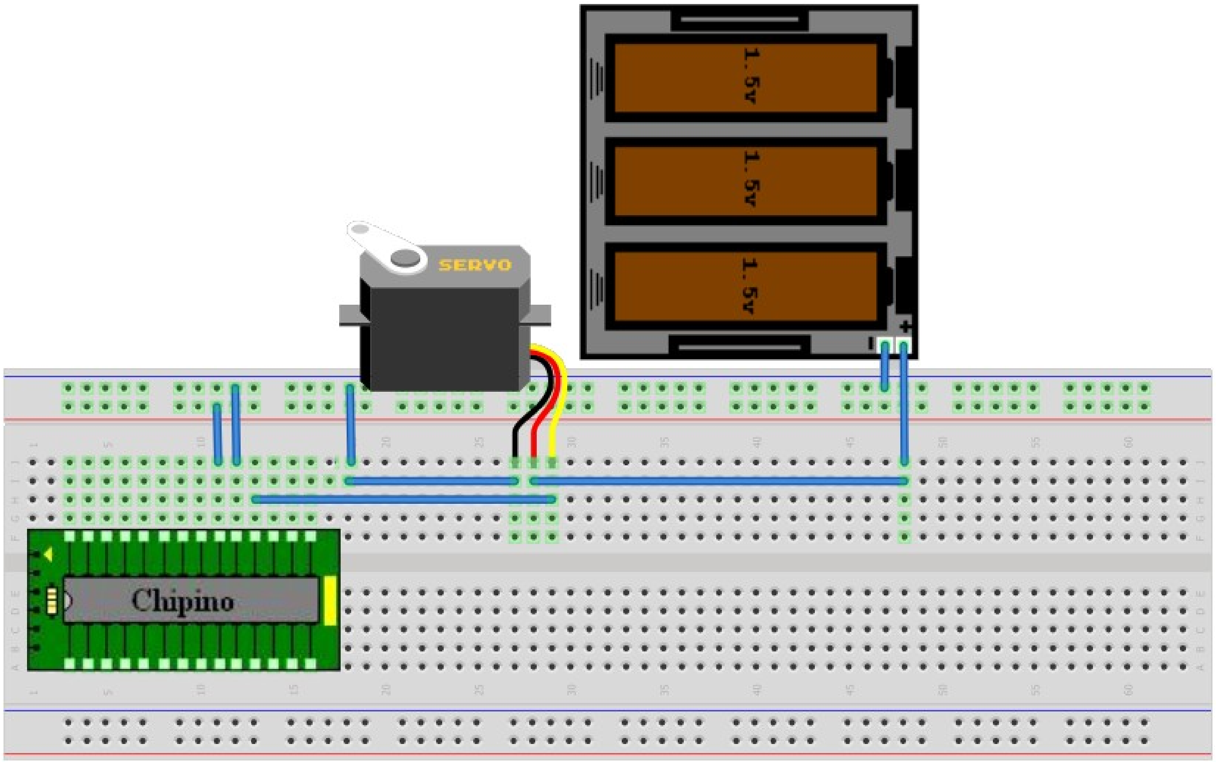
Should be:
CKP = 0;

Beginner’s Guide to Embedded C Programming - Volume 3

Correction #1
Figure 8-2A
Connections to LCD on CHIPINO mini are wrong. The corrected layout is below.



Correction #2
Figure 10-2A
Connections on CHIPINO mini won't work with the Next Steps project.


1. 5v
1. 5v
1. 5v
SERVO
Chipino
-
+