Introduction:

The purpose of the PRESSON Circuit Modules C8051F330GM Microcontroller Development Board is to provide a platform for experimentation and project development using the Silicon Labs C8051F330-GM MCU (microprocessor control unit). In the datasheet for the C8051F330-GM MCU, the manufacturer describes the device as a "system on a chip MCU." This description is quite appropriate because of the rich assortment of peripherals that are present within the IC (integrated circuit) package, operating in support of the embedded 8051 CPU (central processing unit). These peripherals include numerous programmable timers and counters, a 10-bit ADC (analog-to-digital converter), a current-sourcing DAC (digital to analog converter), a UART (universal asynchronous receiver/transmitter), an enhanced SPI™ (serial peripheral interface) port, and a SMB™ (system management bus) port.

PRESSON Circuit Modules, Inc. will provide a comprehensive set of lab/theory exercises covering real world applications of the peripherals contained within the C8051F330-GM MCU. A feature of the C8051F330GM Development Board that makes it particularly desirable for experimentation and project development is the removable C8051F330GM Module. During programming and experimentation using the development board, the module fits securely into a DIP-20 ZIF socket on the development board. When initial programming and experimentation is completed, the user can quickly remove the module and placed it into any standard breadboard or protoboard with a 0.1" grid. If a project requires further
programming revisions or debugging, the user can easily remove the module from the project board and re-inserted into the ZIF socket.

**Printed Circuit Board Specifications:**

The C8051F330GM Development Board circuitry is contained on a two-layer FR-4 epoxy glass PCB (printed circuit board) with 1/2 oz copper laminate. The board dimensions are 4.0” x 2.5.” The lower layer of the board contains a ground plane, with grounded 0.125” mounting holes in the four corners. These holes allow for panel mounting of the PCB, with possible chassis grounding to a metal enclosure or base plate. Figure 1 contains a silk-screen view of the upper layer of the PCB, clearly indicating the location of each component.

![C8051F330GM Development Board Silkscreen](image)

**Figure 1**
Sample pages from the C8051F330GM Microcontroller Development Board User's Assembly Guide:

25) Referring to Table 3 of the Tables and Figures sheets, remove the twenty-position ZIF socket ZIF1 from the C8051F330GM Development Board kit package. Using Figure 13 below as a guide, insert the ZIF socket into the J1 and J2 position on the PCB. Ensure that the locking lever is oriented toward the upper left hand side of the J1 position.

26) Referring to Table 3 of the Tables and Figures sheets, remove the eight-position DIP sockets from the C8051F330GM Development Board kit package. After carefully studying Figure 14 below, insert the socket into the U1 position on the PCB exactly as shown.
27) Secure the DIP socket in place with painter's tape and solder it into position. Also trim the component leads and perform a solder bridge check.

**NOTE**: Capacitors C1, C4, and C5 are not polarized. These multilayer ceramic capacitors (MLCCs) can be positioned in either orientation on the PCB. The first two numbers on the sides of these components form the first two digits of the component's nominal value, based on the picofarad scale. The third digit represents a power of ten by which to multiply this base value. Thus, with 474 on the sides of C1 and C5:

\[
47 \times 10^{-12} F \times 10^4 = 47 \times 10^{-8} F = 0.47 \mu F
\]

28) Referring to Table 3 of the Tables and Figures sheets, remove capacitors C1 and C5 from the C8051F330GM Development Board kit package. Insert the capacitors into position using Figure 15 below as a guide.

![Figure 15](image)

**Figure 15**
Introduction:
The purpose of this lesson is to survey the Data Transfer instructions contained within the CIP-51 Instruction Set. The first part of the theoretical portion of the lesson covers MOV instructions, which comprise the majority of the Data Transfer instruction group. The lesson also covers PUSH and POP instruction, which allow data to be stored in a section of the internal data RAM designated as the stack. The theoretical portion of the lesson ends with a discussion of XCH (exchange) instructions, which involve the swapping of data between the accumulator and a general-purpose register, or between the accumulator and a memory location.

This lesson is a combination of theoretical study and hands-on programming and testing using the Silicon Labs IDE. (It is assumed that the student has completed Lesson 2 and has a foundation knowledge of the Silicon Labs IDE.) After each data transfer instruction is introduced and explained, students will be prompted to perform the corresponding programming exercise, utilizing the Silicon labs IDE and the PRESSON C8051F330GM Development Board. These programming exercises comprise the second part of this lesson. Students are given the opportunity to perform these lab exercise immediately after completion of their corresponding theory sections, or to continue with the theoretical material and perform the programming exercises at a later time.
Objectives:

Upon completion of this lesson, the student should be able to:

- Identify and describe the various MOV (move) instructions contained within the Data Transfer group of the CIP-51 instruction set.
- Identify the three addressing modes used during CIP-51 data transfer operations.
- Identify those MOV instructions within the CIP-51 data transfer group that utilize immediate addressing.
- Identify those MOV instructions within the CIP-51 data transfer group that utilize direct addressing.
- Identify those MOV instructions within the CIP-51 data transfer group that utilize indirect addressing.
- Explain the purpose of the stack within the internal data RAM of the C8051F330 MCU.
- Describe the operation of the Stack Pointer (SP) SFR and explain how it controls access to the stack.
- Describe how PUSH and POP instructions are used to move data to and from the stack.
- Explain the purpose of the Data Pointer (DPTR) and describe its use in accessing the C8051F330 XRAM (external memory).
- Explain how the MOVX instruction is used to access the C8051F330 XRAM.
- Explain how the MOVC instruction is used to access a byte of code data contained in the C8051F330 Program/Data Flash Memory.
- Explain the function of the XCHG (exchange) and XCHGD instructions within the CIP-51 data transfer group.
Programming Exercise 4-1:

1) If the DEBUGADPTR1-USB is no longer connected between your PC and the C8051F330GM Development Board, then complete the connection. Refer to Part 1 of Lesson 2 as necessary to ensure you are making the connection properly.

Until the steps involved in creating a project become second nature, it is strongly recommended that you keep the Procedure for Creating a Project (Part 2 of Lesson 2) within view. This and subsequent programming exercises will refer to steps contained within that procedure.

2) Reopen the Silicon Labs IDE as necessary and perform the following steps:
   - Open the ASM_Source_Code folder, located within your My_C8051F330GM_Proj folder. Locate the Reg_MOV text file. Open the file and save it as a .asm file.
   - Open a new project within your My_C8051F330GM_Proj folder.
   - Name the project MOV_Demo. (The project will contain exercises demonstrating basic register-to-register, direct, indirect, and immediate data transfer using the MOV instruction.)
   - Select the C8051F330_5 device family and specify the project type as an ASM Source Project.
   - As illustrated in Figure 4-16a, right click on the Source Files Icon and select Add files to group Source Files. The Contents of the ASM_Source_Code folder should now appear.
   - Scroll through the listing and locate the Reg_MOV.asm file.
   - Double click on this file to add it to the project. The Project window should now appear as seen in Figure 4-16b.
   - Right click on the Reg_MOV.asm icon and select Add Reg_MOV.asm to build. The Project window should now appear as seen in Figure 6-16c.
   - Double click on the Reg_MOV.asm icon. This program should now appear in the Debug window.
   - Assemble and build the project.
   - Download the code to the development board. (You might need to open the Options window to respecify the Serial Adapter as USB Debug Adapter and the interface as C2.)
3) Open the 8051 and Disassembly windows of the IDE. Then adjust the Debug, 8051, and Disassembly windows for optimal viewing (as represented in Figure 4-17). Note the program contained in Figure 4-17 is the same as that discussed in Example 4-1. You are advised to review that example as necessary before performing the remaining program steps.

4) Click on the RESET button to bring the code shown in the Disassembly window to program flash memory address 0000.