DIP SWITCH

Revision 04.05.04

Class

Instructor / Professor

CONTENTS

1 REQUIREMENTS

1.1 Hardware

To complete this lab, **the following hardware is required**:

- Axiom CML-12C32 Development Kit
- PC running Windows OS
- Project Starter Kit which includes:
    - (1) Four position DIP switch
    - (4) 10 k ohm resistors ¼ w
    - (10) Jumper Wires

1.2 Software

The CML-12C32 board used in this experiment comes with all the software needed to complete this project.

There are many additional utilities included on the boards support CD that can make developing your own projects easier. The CD contains example source code, microcontroller and board documentation and experiments for all Axiom development boards. You can even download the latest versions of the software and documentation free from our web site at: www.axman.com.

Also included is an integrated development environment, called AxIDE, for communicating with the board (via the serial port) and for reading and writing its flash memory. To complete this Lab, you should have this program installed on a PC running Microsoft Windows (95/98/2000/XP).

NOTE: This lab does not teach you how to use the AxIDE terminal interface or the MON12 Monitor program to modify memory and upload programs. It assumes you're already familiar with these procedures. Refer to your board manual for details on installing and using this software, including a tutorial for using AxIDE.

CAUTION

Devices used in this lab are static sensitive and easily damaged by mishandling. Use caution when installing wires and devices onto the board to prevent bending the leads.

Experiments should be laid out in an orderly fashion. Start your lab time with the bench clean and free of metal objects and leave the lab area in a clean condition by picking up loose parts, wires and small objects.

2 GETTING STARTED

This lab will show you how to add a DIP switch as an input device to the microcontroller on your Axiom development board. In this example, a four position DIP switch is used.

A four position DIP switch has four SPST switches inside. Each switch is open or closed depending on the position of the switch lever.

In this lab, an analog port on the HC12 is used as a digital input. A digital level can be read from this port which will be high or low depending on the position of the switch. A resistor connects each of the outputs of the switches to +5v, which pulls the signal high. Closing the switch will connect the output to ground.

A DIP switch is a good input device for appliances, machinery, cars, & alarms plus many others. The user can select different options or operating modes by just moving the position of the switch. They come in several sizes such as two, four and eight position.

3 LAB PROCEDURE

This lab is arranged in a series of steps. Each step should be completed before moving on to the next one, which builds on prior ones. Repeat each step as many times as necessary to become familiar with it. You will find it easier to complete more complex experiments after mastering the simple ones.

As an aid to keeping track of location it's a good idea to mark each step as it's completed, since the experiment will fail if anything is skipped.

3.1 Description

This example uses PORT AD on the HC12SC32 microcontroller. This analog port has a digital input register. The output of the DIP switch is connected directly to these inputs, so reading the level on port pins 4 - 7 will return the level on the DIP switch, whether high or low. This port is enabled for digital input by writing a one to each bit in the ATDDIEN register ($008D). Since we are using bits 4 – 7, then a $F0 should be written to this register. Reading register PTAD ($0270) will return the level on the dip switch.

3.2 Detailed Steps

This section describes how to build the DIP Switch project and test it with the monitor running on the CML-12C32. In the next section, you'll see how to write a simple program that uses this switch.

> **NOTE**: To complete these steps you must be familiar with modifying register contents on your board. For example, to write $F0 to the ATD0DIEN register with the MON12 monitor, type `MM 8D` at the monitor prompt and press <enter>. Then type `F0` <enter> then (`.`) to end modify mode. Type Help for more commands.
>
> You can use a different monitor or debugger if you prefer, such as the GNU GDB.

1. Verify power is NOT applied to development board.

2. Install the DIP switch and the 4 resistors on the breadboard area as shown in **Figure 1**.

3. Install the jumper wires on the board as follows:

MCU_PORT	**Breadboard**	**DIP Switch**	**Breadboard**
PAD4	D1	G1	GND
PAD5	D2	G2	GND
PAD6	D3	G3	GND
PAD7	D4	G3	GND
+5V	+5V		
GND	GND		

4. Configure the CML-12C32 board to start the MON12 debugger by setting the jumpers to their default positions (NO_AUTO and MEM_EN are ON)

5. Apply power to the board.

6. Turn all switches on the DIP switch to ON.

7. Using the debugger software, write $F0 to ATDDIEN. This configures the analog port bits 4 – 7 as digital inputs for reading the outputs of the DIP switch.

8. Read PTAD and verify a reading of $0x. X means don't care and 0 is the reading when the all switches are on and the output is connected to ground.

9. Turn DIP switch position 1 OFF then read PTAD and verify a reading of $1x. 1 is the reading when PAD0 is high.

10. Turn DIP switch position 1 ON and position 2 OFF. Read PTAD and verify a reading of $2x. 2 is the reading when PAD1 is high.

11. Turn DIP switch position 2 ON and position 3 OFF. Read PTAD and verify a reading of $4x. 4 is the reading when PAD2 is high.

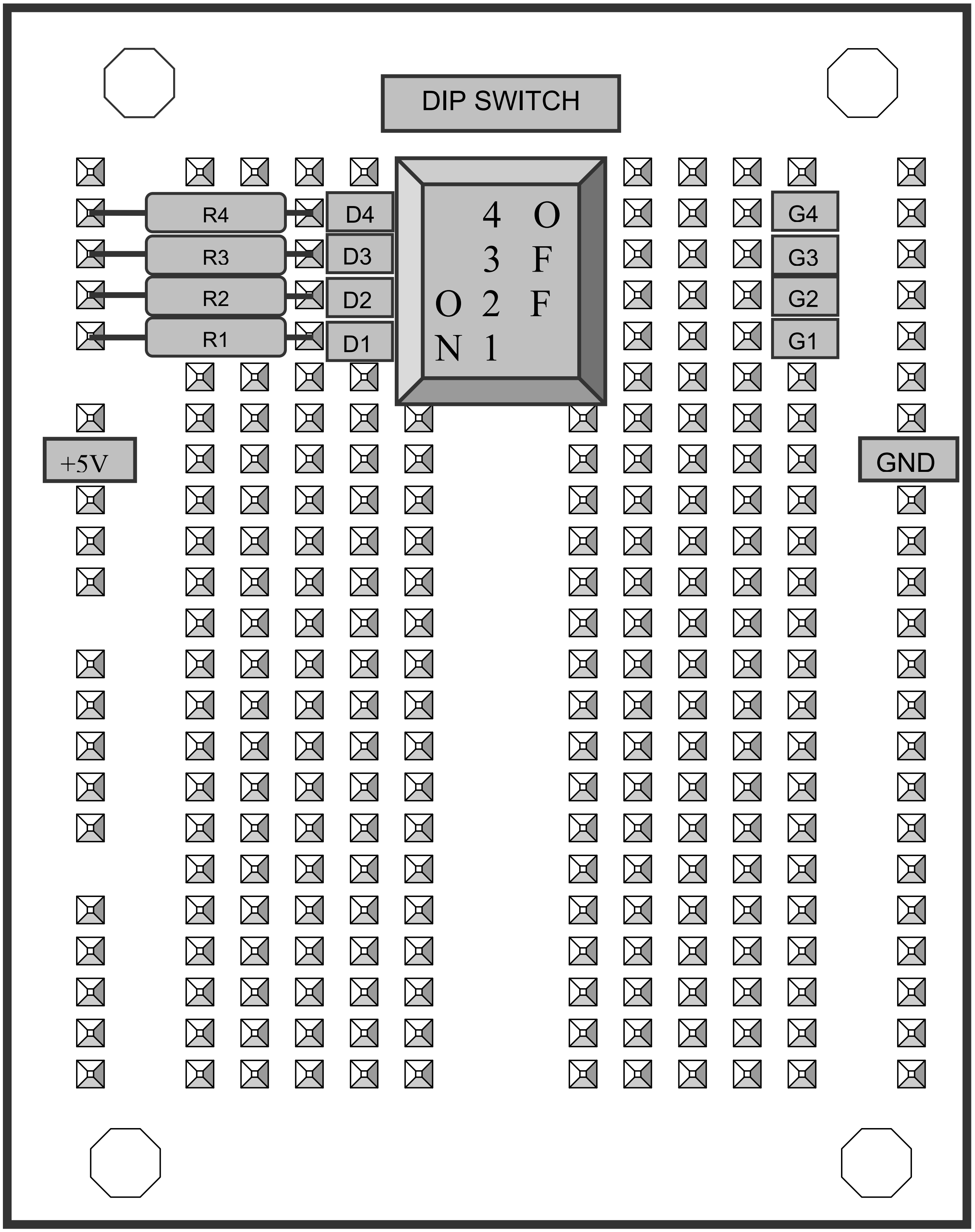
12. Turn DIP switch position 3 ON and position 4 OFF. Read PTAD and verify a reading of $8x. 8 is the reading when PAD3 is high.

13. Turn all DIP switch positions OFF. Read PTAD and verify a reading of $Fx. F is the reading when all DIP switches are high on the output.

These steps show how easy a DIP Switch can be added to a microcontroller. By using four bits of the analog port as digital inputs, the level of the DIP switch is read. Any combination of DIP switches can be set ON or OFF.

Your software application can assign each switch a function. This can change how the hardware behaves based on the switch settings. This allows the same hardware to be used in different devices by just changing the switch setting.

Figure 1



4 DIP SWITCH PROGRAM

The previous section described a method of reading the switch manually using a debugger. While this method is useful for testing and experimenting, once the hardware is working you'll want to write a software program to read the switch.

This section will describe how to write such a program in assembly language. The full source code listing is at the end of this section. Both source code and assembled executable for this example can also be downloaded from the Axiom web site: **www.axman.com**.

If viewing this document on your PC, you can copy and paste the source code into a text editor (such as notepad) then save and assemble it using AxIDE.

Refer to the owner's manual of your board for instructions on creating software and running programs for your development board.

4.1 Program Description

The program first sets bits 4 - 7 of the Analog Port to digital input mode. A subroutine is called that reads the DIP switch thru the Analog Port and displays the level as a 0 (logic low) or 1 (logic high) on the terminal. A short delay is called between loops of the program. By changing any of the four DIP switches, you can see the level change on the terminal.

4.2 Running the Program

1. Upload the assembled program DIPCD.S19 into the RAM on your board. This program starts at address $0800, which is internal memory. The source code for this program is shown below.
2. Execute the program by typing `call 0800` in the terminal and pressing <enter>.
3. The terminal will display the level of each switch as a 0 or 1.
4. Turn the DIP switches ON or OFF and you will notice the level is updated on the terminal.
5. Try several combinations, and verify that the display follows the position of the DIP switch.
6. Press the reset button on the board or remove power to stop the program.

4.3 DIP Switch Program Source Code

```
*        DIP Switch Display Program
*        HC12 code
*
;
ATDDIEN:   equ     $008D        ; Digital/Analog configure
PTAD:      equ     $0270        ; Digital Input of Analog Port
DIP1:      equ     $01          ; DIP Switch position one
DIP2:      equ     $02          ; DIP Switch position two
DIP3:      equ     $04          ; DIP Switch position three
DIP4:      equ     $08          ; DIP Switch position four
SCISR1:    equ     $00CC        ; Sci status register
SCIDRL:    equ     $00CF        ; Sci data register

          org      $0800        ; start address
          movb     #$F0,ATDDIEN  ; Configure for digital read
Loop1
          ldaa     PTAD       ; get DIP switch value
          bsr      OutBin     ; shift data and send
          bsr      Delay      ; Delay for display
          bra      Loop1

; Read DIP Switch and mask bits
Read:
          ldaa    PTAD        ; Read DIP Switch
          anda    #$F0        ; get only bits 0 - 3
          rts

; Output A to SCI0
Output:
          brclr  SCISR1,#$80,Output  ; wait for Xtrans to empty
          staa   SCIDRL       ; send character
          rts

; Output Acc A has 4 binary digits in AscII format
OutBin:
          bsr    CrLf         ; new line
          lsra                ; ready A for display
          lsra                ; ready A for display
          lsra                ; ready A for display
          bsr    BitShift     ; send first bit
          bsr    BitShift     ; send second bit
          bsr    BitShift     ; send third bit
          bsr    BitShift     ; send fourth bit
          rts

; Shift bit for display
BitShift:
          lsra                ; next bit
          psha                ; save A
          anda   #$01         ; mask bit
          adda   #$30         ; add AscII offset
          bsr    Output       ; send bit
          pula                ; restore A
          rts
```

```
; send CR & line feed
CrLf:
          psha                    ; save A
          ldaa     #$0a           ; send cr
          bsr      Output
          bsr      Delay
          ldaa     #$0d           ; send line feed
          bsr      Output
          bsr      Delay
          pula                    ; restore A
          rts
Delay:
          ldab     #$20
DelayB
          ldy      #$8000
DelayA:
          dey
          bne      DelayA
          decb
          bne      DelayB
          rts
```

5 QUIZ

Answer the following questions based on the example presented in this lab.

1. Where is the program DIPCD.S19 located in memory?
A. External memory **B.** Internal memory **C.** Eprom **D.** Rom

2. What is the main point of writing $F0 to ATDDIEN?
A. Change Level **B.** Addition **C.** Digital Input **D.** Analog Input

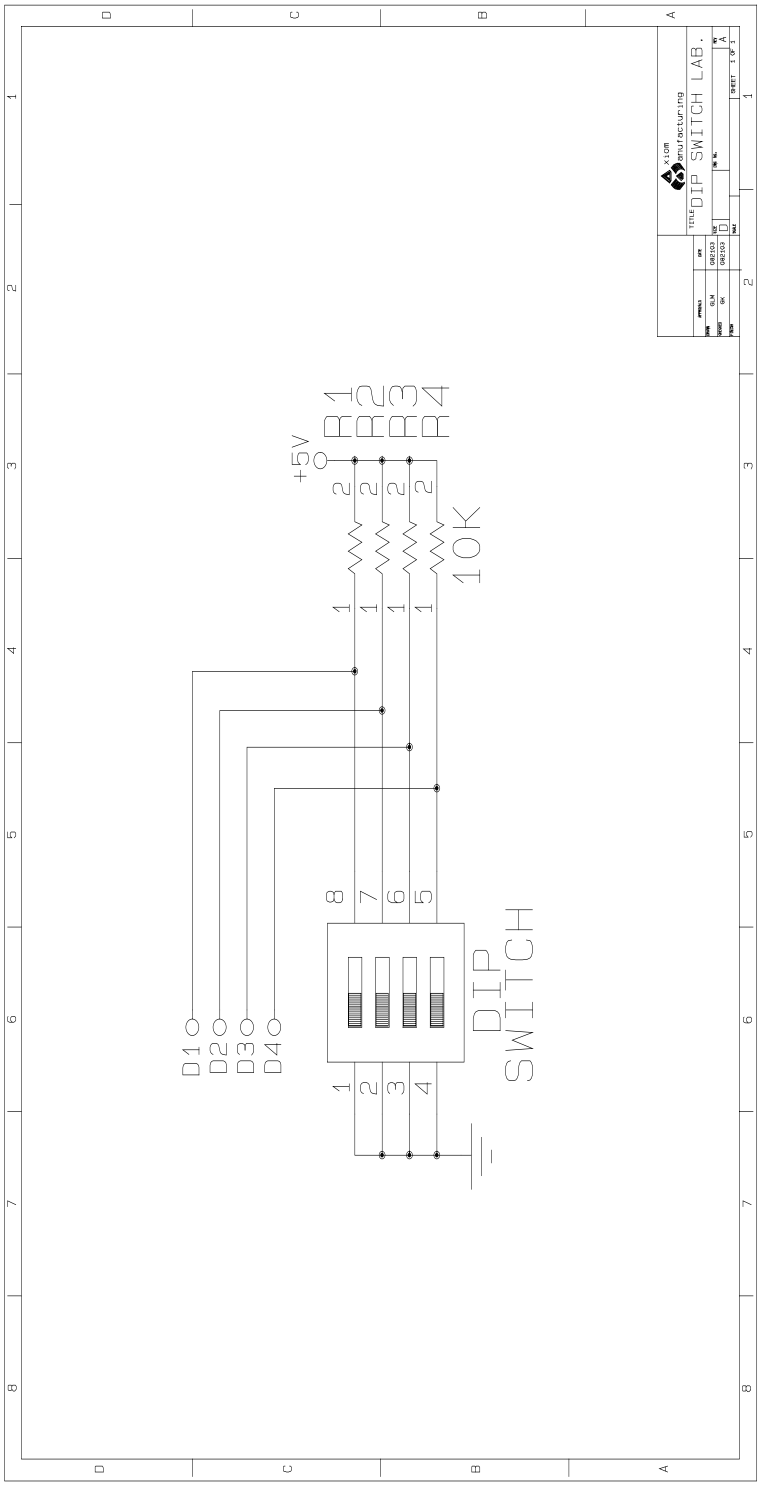
3. The DIP switch contains four what?
A. SPST Switches **B.** Resistors **C.** DPDT Switches **D.** Diodes

4. The resistors are used as?
A. Pull-Down **B.** Bias **C.** Pull-Up **D.** Disable

5. PTAD can be used to read?
A. Analog **B.** Digital **C.** Analog/Digital **D.** Pulse Width

BONUS QUESTION: How are DIP switches used with a microcontroller?
A. Change Mode **B.** Input Device **C.** Allow multi-functional devices **D.** All of the above

6 SCHEMATIC


+5V
R1
R2
R3
R4
10K
DIP SWITCH
D1
D2
D3
D4
Axiom Manufacturing
TITLE DIP SWITCH LAB.
REV A
SHEET 1 OF 1