

# *DIP SWITCH*

Revision 03.11.13

---

Class

---

Instructor / Professor

## LICENSE

You may use, copy, modify and distribute this document freely as long as you include this license and the Axiom Manufacturing copyright at the bottom of this page. You can download the latest version of this document from the Axiom website: [www.axman.com](http://www.axman.com)

# CONTENTS

<b>1</b>	<b>REQUIREMENTS.....</b>	<b>3</b>
	1.1 HARDWARE .....	3
	1.2 SOFTWARE .....	3
<b>2</b>	<b>GETTING STARTED.....</b>	<b>4</b>
<b>3</b>	<b>LAB PROCEDURE .....</b>	<b>4</b>
	3.1 DESCRIPTION.....	4
	3.2 DETAILED STEPS.....	4
<b>4</b>	<b>DIP SWITCH PROGRAM.....</b>	<b>7</b>
	4.1 PROGRAM DESCRIPTION .....	7
	4.2 RUNNING THE PROGRAM.....	7
	4.3 DIP SWITCH PROGRAM SOURCE CODE .....	8
<b>5</b>	<b>QUIZ.....</b>	<b>9</b>
<b>6</b>	<b>SCHEMATIC .....</b>	<b>10</b>

# 1 REQUIREMENTS

## 1.1 Hardware

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

- Axiom CME11E9 EVBU Development Kit
- PC running Windows OS
- DIP SWITCH Lab Kit which includes:
  - (1) Four position DIP switch
  - (4) 10 k ohm resistors  $\frac{1}{4}$  w
  - (10) Jumper Wires

## 1.2 Software

The CME11E9-EVBU 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, 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](http://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 Buffalo 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 HC11 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 PORTE on the HC11E9 microcontroller. This analog port has a digital input register. Reading the level on port pins PE4, PE5, PE6 and PE7 will return the level on the DIP switch, whether high or low.

### 3.2 Detailed Steps

This section describes how to build the DIP Switch project and test it with the monitor running on the CME11E9-EVBU. 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 reading register contents on your board. For example, to read PORTE with the buffalo monitor, type MD 100A into the terminal at the buffalo prompt and press <enter>. The value will be the byte in the 11<sup>th</sup> column on the first row. 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:

**EVBU P4 ---- Breadboard**

GND-----GND(-)  
 VCC-----VCC(+)  
 PE4 -----D1  
 PE5 -----D2  
 PE6 -----D3  
 PE7 -----D4

**Breadboard --- Breadboard**

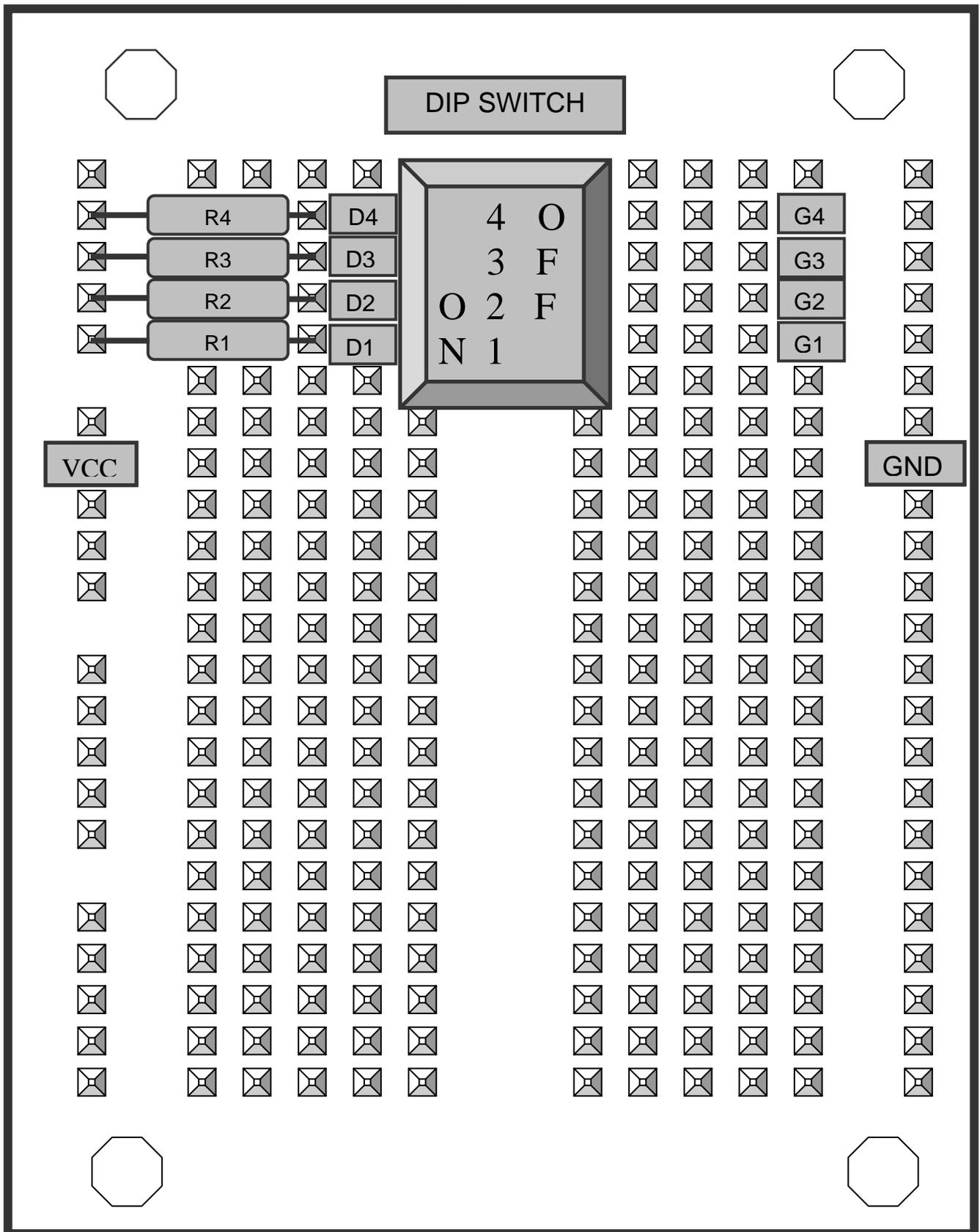
G1 ----- GND(-)  
 G2 ----- GND(-)  
 G3 ----- GND(-)

4. Install the MODA jumper on the EVBU board and remove jumpers MODB and MEM\_EN. This will configure the board for single chip operation.
5. Apply power to the board.
6. Turn all switches on the DIP switch to ON.
7. Using the monitor, read PORTE 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.
8. Turn DIP switch position one OFF then read PORTE and verify a reading of \$1x. 1 is the reading when PE4 is high.
9. Turn DIP switch position one ON and position two OFF. Read PORTE and verify a reading of \$2x. 2 is the reading when PE5 is high.
10. Turn DIP switch position two ON and position three OFF. Read PORTE and verify a reading of \$4x. 4 is the reading when PE6 is high.
11. Turn DIP switch position three ON and position four OFF. Read PORTE and verify a reading of \$8x. 8 is the reading when PE7 is high.
12. Turn DIP switch position one, two, three, four OFF. Read PORTE 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 port E as inputs, the level of the DIP switch is read. Any combination of DIP switches can be set ON or OFF in any combination.

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](http://www.axman.com).

If viewing this on your PC, you can also copy and paste the source code below 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 reads PORTE bits 4 thru 7. Each of the channels is displayed 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 DIP1.S19 into the RAM on your board. This program starts at address \$0100, which is internal memory. The source code for this program is shown below.
2. Execute the program by typing `call 100` in the terminal and pressing <enter>.
3. The terminal will display the level of each switch as a 0 or 1.
4. Turn one of 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
*      HC11 code
*
PORTE:   equ      $100A      ; PORT E input resister
OUTPUT:  equ      $FFAF      ; Buffalo output routine
OUTCRLF: equ      $FFC4      ; Buffalo routine CR/LF
OUTRHLF: equ      $FFB5      ; Buffalo routine right nibble to AscII

                org      $0100      ; start address

Loop1
        bsr      OutBin      ; shift data and send
        bsr      Delay      ; Delay for display
        bra      Loop1

* Output Acc A has 4 binary digits in AscII format
OutBin:
        jsr      OUTCRLF      ; new line, buffalo routine
        ldaa     PORTE        ; read DIP switch
        lsra
        lsra      ; shift bits
        lsra      ; into position
        lsra      ; for display
        lsra
        lsra
        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
        jsr      OUTRHLF     ; send right nibble has AscII
        pula      ; restore A
        rts

* Delay
Delay:
        ldab     #$04

DelayB
        ldy      #$FFFF

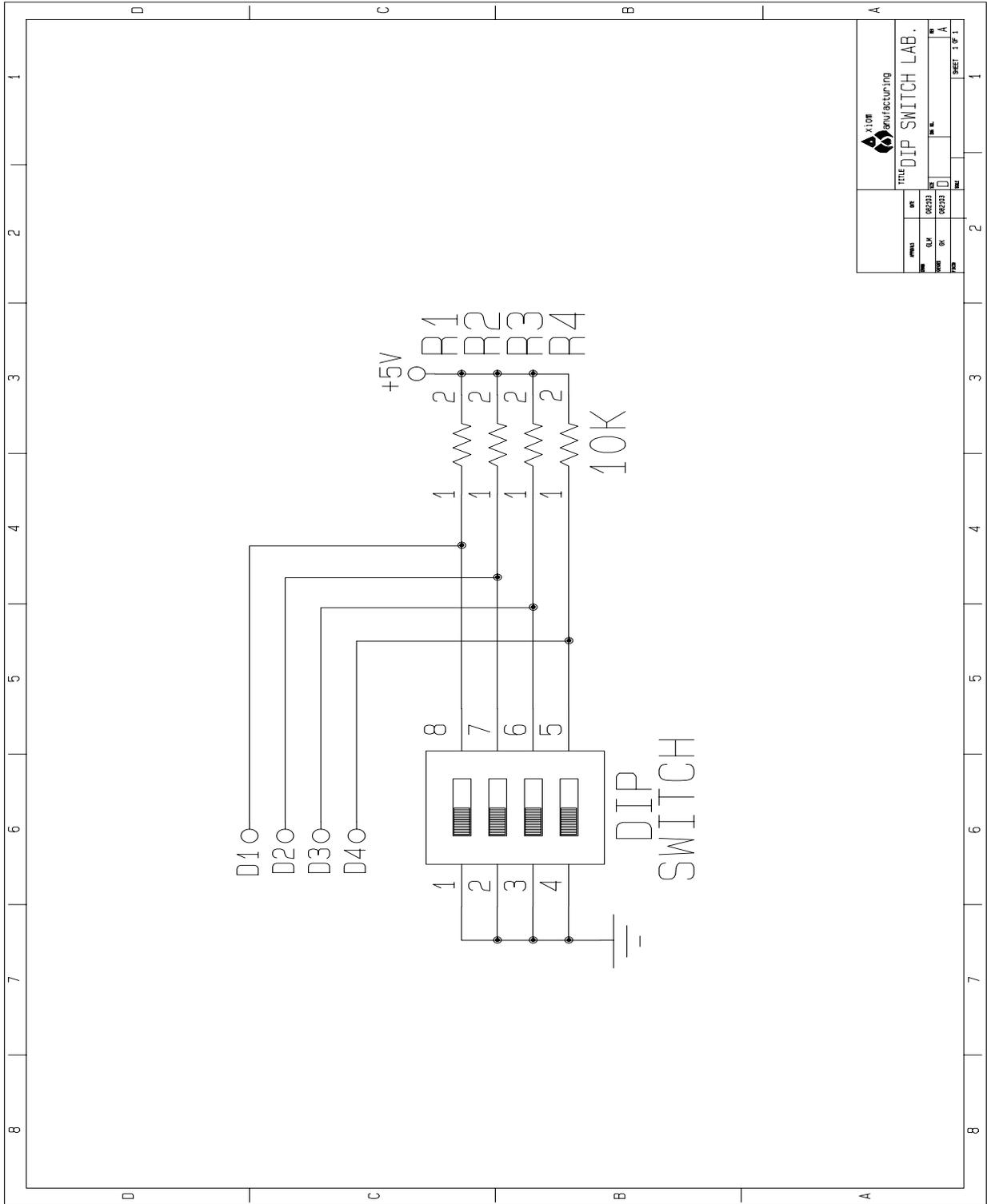
DelayA:
        dey
        bne     DelayA
        decb
        bne     DelayB
        rts
```

# 5 QUIZ

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

1. Where is the program DIP1D.S19 located in memory?  
A. External memory    B. Internal memory    C. Eeprom    D. Rom
  2. What register is used in reading the DIP switch?  
A. PORTA    B. PORTB    C. PORTE    D. PORTF
  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. PORT E 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



REV		DATE		BY		CHK		APP		SHEET 1 OF 1	
 TITLE: DIP SWITCH LAB.											