POTENTIOMETER

Revision 03.11.13

____

Class

____

Instructor / Professor

2813 Industrial Lane - Garland, TX 75041

Phone: 972-926-9303
http://www.axman.com
email: sales@axman.com

CONTENTS

1 REQUIREMENTS

1.1 Hardware

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

- Axiom CME11E9 EVBU Development Kit
- PC running Windows OS
- Seven Segment Lab Kit which includes:
    - (1) 10k Potentiometer 20-22 turns
    - (1) 10k Resistor
    - (3) 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.

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 an external potentiometer control device to your Axiom development board.

In this lab, a 10 k potentiometer (POT) is used with Port E on the microcontroller. On the HC11, Port E is an eight channel analog to digital (A/D) port, which is required for reading the voltage on the POT. Channels on this A/D port are selected by an analog mux. The A/D requires a reference which, on the EVBU board, is connected to +5v. The potentiometer is connected between +5 and GND.

This experiment will use the center tap on the potentiometer connected to the analog port of the microcontroller, which is configured to continuously convert data on channels 4 thru 7. The POT can be manually adjusted from +5 to GND level. By adjusting the POT in steps, the analog port can create a hex value. The results of this A/D conversion can be read by software from ADR4 as a number from 0 to $FF.

Potentiometers are commonly used throughout the electronics industry as input devices and sensors in many products such as appliances, machinery, cars and robotics. They come in many shapes and sizes for board or chassis mounting.

3 LAB PROCEDURE

The 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 must first be configured to continuously scan for analog input. After the potentiometer is adjusted all the way to GND, a value is read from the analog port which should be close to $00.

Manually adjusting the POT will cause this input value to change. A visual picture of the POT control will be created by reading the input value of each POT adjustment and marking it on a graph.

3.2 Detailed Steps

This section describes how to build the potentiometer 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 device.

> **NOTE**: To complete these steps you must be familiar with reading and writing 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 11th column on the first row. Type Help for more commands.

1. Verify power is NOT applied to development board.

2. Install a potentiometer and 10k resistor on the breadboard area as shown in **Figure 1**.

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

   EVBU P4 ---- Breadboard
   GND----------GND
   PE7 -----------TAP
   VCC ----------FULL

4. Install the MODA jumper on the EVBU board and remove jumpers MODB and MEM_EN to configure the board for single chip operation then apply power to the board.

3.2.1 Monitoring the Potentiometer

Register address quick reference (see M68HC11E Technical Data manual for more)

OPTION	$1039	ADR1	$1031	ADR3	$1033
ADCTL	$1030	ADR2	$1032	ADR4	$1034

5. Using the monitor and terminal, write $80 to the OPTION register. This sets the ADPU bit high which turns the analog to digital converter on. Note: the buffalo monitor turns this on for you because it must be written in the first 64 cycles after reset.

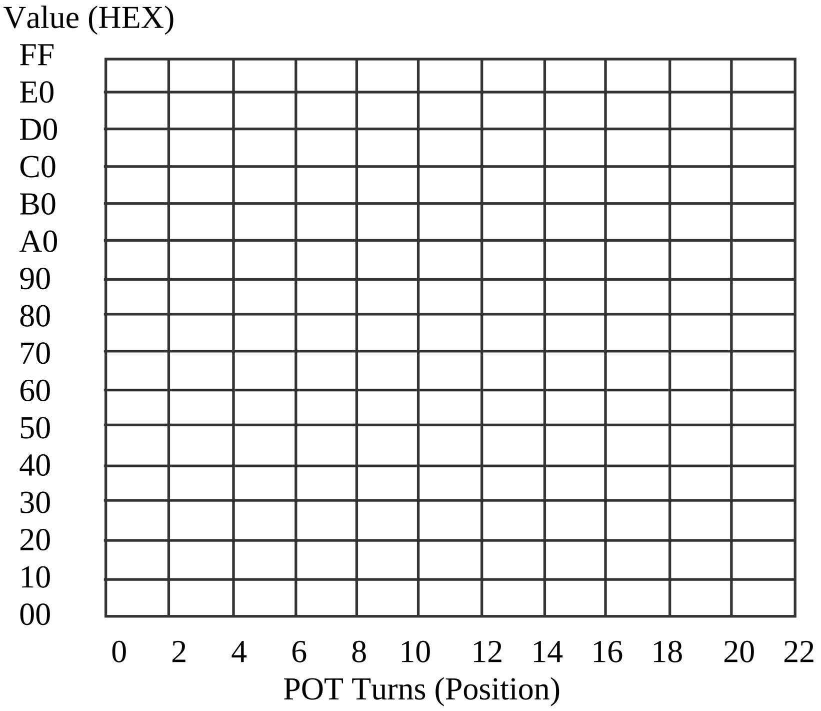
6. Write $34 to the ADCTRL register. This sets the A/D converter to continuous conversion on multiple channels. It also selects port E channels 4,5,6,7. The conversion results will be placed in ADR1 - ADR4.

7. Adjust the POT and read ADR4 until you are able to read a value as close to $00 as the POT adjustment will allow. Record this value in the graph on the following page at 0 turns.

8. Adjust the POT two full turns in the opposite direction. Read and record the ADR4 value on the graph at 2 turns

9. Continue adjusting the POT two full turns while recording the ADR4 value on the graph until reaching 22 full turns.

10. Plot the result on the graph by drawing a line between the points recorded.



3.2.2 Limiting Range

11. Change the jumper wires on the board as follows:

EVBU P4 ---- Breadboard
GND----------GND
PE7 -----------TAP
VCC ----------HALF

12. Repeat steps 7-10 above to plot the adjusted values.

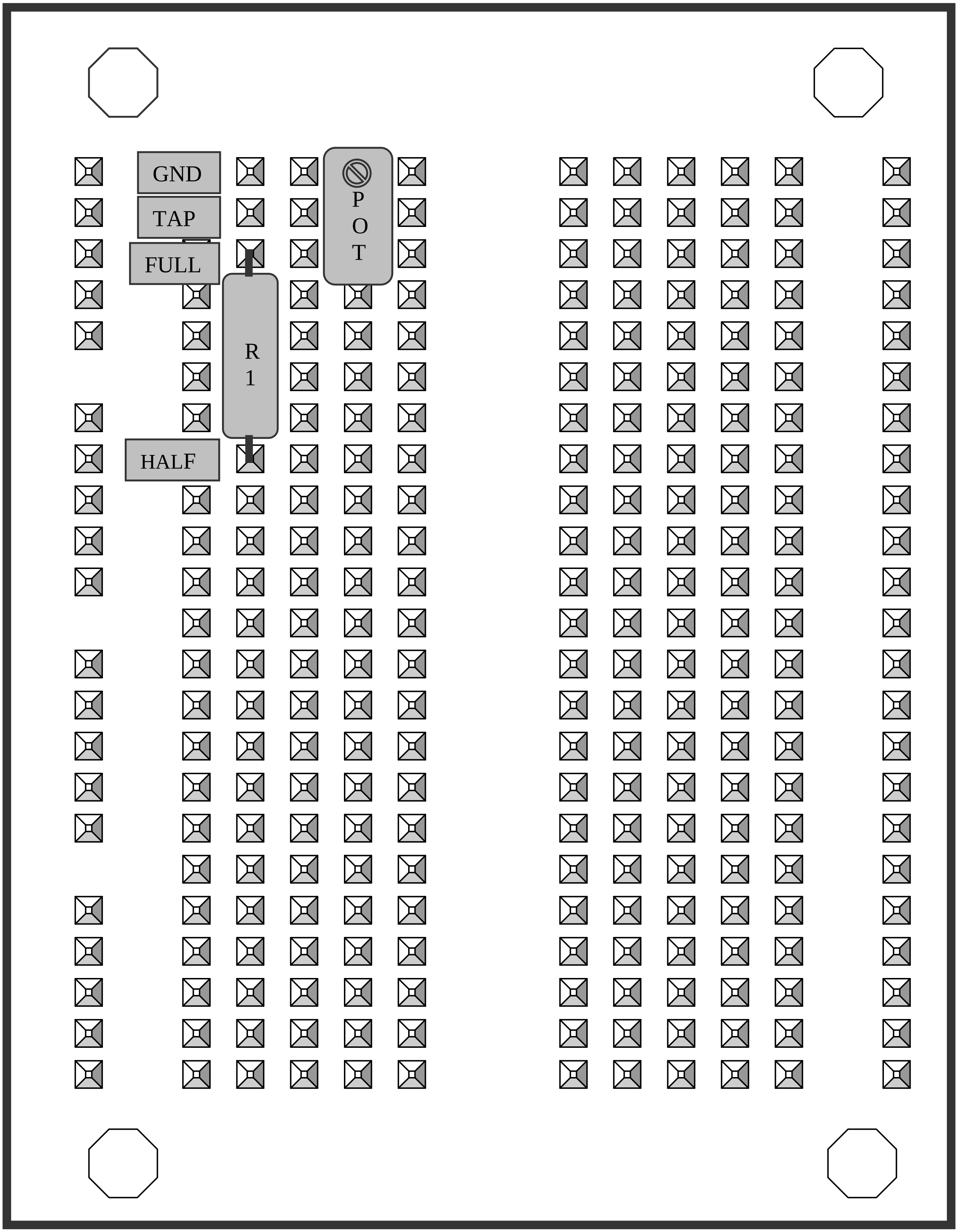
As seen in the graph, you can use an analog port to determine the position of a potentiometer connected to a microcontroller.

By adding a 10k resistor in series with the +5 side of the POT, the range of adjustment can be reduced by one half. This is helpful in cases where the pot is driven by a higher voltage than the microcontroller input voltage limit.

The range and position of the POT on the graph can also be adjusted by adding a resistor to GND, +5 or both sides of the POT. The normal analog channel input range is 0 to 5 volts. With a 10k resistor, this can be extended from 0 to 10 volts in range.

Caution should be excised not to exceed the input range of the analog input. The maximum voltage on the HC11 power supply is +5 volts.

Figure 1



4 POTENTIOMETER PROGRAM

The previous section described a method of reading the potentiometer level 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 use this device.

This section describes 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 be downloaded from the Axiom web site: **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

This program will display the results of our A/D conversion on the potentiometer position on the serial terminal. First the program must turn on and configure the A/D channel for multi channel conversion on channels 4 thru 7 and set it for continuous conversion mode.

To save code, this example uses the Buffalo monitor routine "OUTCRLF" which is part of the monitor. This is called to output a carriage return and line feed for allowing the next value to be displayed. Register X is loaded with the address of ADR4 which contains the A/D conversion results. Another Buffalo routine is then called to display this value as two hex numbers on the terminal.

Finally, a delay is called to slow the display down for the operator and terminal. The program then jumps back to the beginning and repeats forever.

4.2 Running the Program

1. Upload the assembled program POT1A.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 result of the A/D conversion as it is continually read. By adjusting the potentiometer, you can see the changes in the terminal window.

4. Press the reset button on the board or remove power to stop the program.

4.3 Potentiometer Program Source Code

```
*     Potentiometer Example Program for the 68HC11
*     This example uses serial output routines in the Buffalo Monitor
*
OPTION       equ     $1039        * A/D On
ADCTL        equ     $1030        * A/D control
ADR4         equ     $1034        * A/D result

* Buffalo routines
OUTCRLF      equ     $FFC4        * car/line feed
OUT1BYT      equ     $FFBB        * display byte as hex

* This subroutine will turn on the A/D convertor
             org     $0100
             ldaa    #$80         * turn A/D on
             staa    OPTION
             ldaa    #$34         * A/D control
             staa    ADCTL        *

* read analog value and display

PotLoop
             jsr     OUTCRLF      * send car/LF
             ldx     #ADR4        * point to analog port
             jsr     OUT1BYT      * send hex value
             bsr     Delay        * delay
             bra     PotLoop
Delay
             ldy     #$FFFF       * delay value
Delay1
             dey                  * decrement count
             bne     Delay1       * repeat until done
             rts                  * return
```

5 QUIZ

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

1. What type port is required for reading a potentiometer?
A. Digital to Analog **B**. Output **C**. Analog to Digital **D**. Serial

2. What is the maximum voltage allowed on port E?
A. 5 volts **B**. No Maximum **C**. 2.5 volts **D**. 10 volts

3. What type of data is read from the A/D?
A. Decimal **B**. Hex **C**. Sine **D**. Matrix

4. What address is OPTION register located?
A. $1039 **B**. $1030 **C**. $0100 to $01FF **D**. External

5. What is the range with the 10k resistor installed in the 5v side?
A. 2X **B**. 1/2X **C**. 4X **D**. 1X

6. Using the completed graph, what can one determine from a value on the graph?
A. Position **B**. Voltage **C**. Turns **D**. All of the above

7. In this experiment, the A/D is setup for ______ conversion?
A. Digital **B**. Single **C**. Continuous **D**. One shot

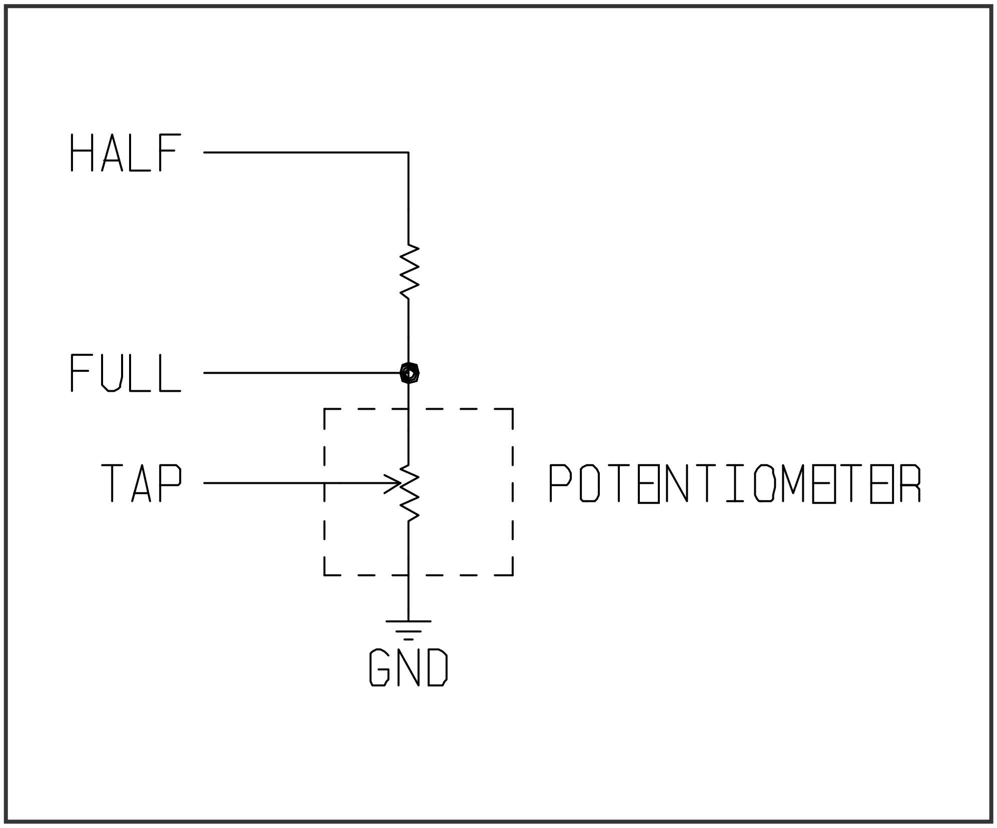
8. How many A/D channels are on the HC11?
A. 8 **B**. 3 **C**. 2 **D**. 16

9. The A/D channels require what?
A. Computer **B**. Resistor **C**. Diode **D**. Reference

10. How many analog channels are converted in multiple channel mode?
A. 1 **B**. 5 **C**. 8 **D**. 4

BONUS QUESTION: The eight channel analog input on the HC11 goes though a?
A. Analog Mux **B**. Switch **C**. Digital Mux **D**. Decimal conversion

6 SCHEMATIC


HALF
FULL
TAP
POTENTIOMETER
GND