POTENTIOMETER

CML12S LAB EXPERIMENT

I

_____

Class

_____

Instructor / Professor

2813 Industrial Ln. • Garland, TX 75041 • (972) 926-9303 FAX (972) 926-6063
support@axman.com Rev 1.00 • web: http://www.axman.com

CONTENTS

1 GETTING STARTED

The following section has been designed to help users to quickly learn proper setup and operation of the lab experiment.

1.1 Introduction

The experiment requires a single board development system that is fully assembled and fully functional from Axiom Manufacturing. Development board CML12S is supported in this

experiment. The manual comes complete with necessary drawings and instructions. All software and drawings are contained on the Axiom Manufacturing CD.

1.2 Software

The CD comes with AxIDE, which is an integrated development environment designed exclusively for use with Axiom development boards, providing an interface to programs running on these boards. AxIDE also makes uploading programs easy via the COM port. Read your board manual for setting up AxIDE.

1.3 Support Software

There are many useful programs on the included CD that can make developing projects easier. The CD also contains example software programs for this experiment on each board. You can also download the latest software free from our web site at: http://www.axman.com.

1.4 Hardware

The following hardware is required:

Axiom CML12S
Windows based PC
10k Potentiometer 20-22 turns
10k ½ w Resistor

2.0 Visual

Devices used in this lab are static sensitive and are easy damaged by mishandling. Use caution when installing wires and devices on the breadboard to prevent the bending of leads. Experiments should be laid out in a orderly fashion. Start your lab time with the bench clean and free of metal objects. Leave the lab area in a clean condition by picking up loose parts, wires and small objects.

3.0 Theory

Axiom's development board is designed for quickly and effectively learning the basics of microcontrollers. This lab will walk the student though the steps of using the development board for its intended purpose, controlling devices. A potentiometer is one device that is commonly used by a microcontroller. In this lab, a 10 k potentiometer is used for the experiment using the analog port channel 7 (PAD7). The analog port is a eight channel analog to digital (A/D for short) port, which is required for reading the voltage on the potentiometer. The channels are selected by a analog mux. The A/D requires a reference. On the CML12S, the reference is connected to +5v. The potentiometer is connected between +5 and gnd. You will be using the center tap on the potentiometer connected to the analog port of the microcontroller. The device can be adjusted from the +5 level down to gnd level. By adjusting the pot in steps, the analog port will be read out in a hex value. The analog port is set up for

single conversion of channels 0 thru 7. The results of this experiment are read from ADR07H at address $009E and $009F. The value from the A/D is read as a hex value with a range of 0 to $03FF. Potentiometers are used in appliances, machinery, cars, servo positioning & robots plus many others. They come in many shapes and sizes for board mounting or chassis mounting.

4.0 Procedure

The procedure is arranged in a series of steps. Each step is to be completed before moving on to the next step. As each step is built on prior steps, the student will increase their knowledge for other labs or self-study. You should go though the steps as many times as necessary to master the subject. As an aid in keeping track of location, the check box next to each step should be checked as completed.

4.1 Description:

You will be using analog port 7 (PAD7) on the DP256 microcontroller. First the analog port must be setup for single scan of analog input. Next the potentiometer is adjusted all the way to the gnd end of the potentiometer. The value is read from the analog port, which should read close to $0000. The value determines the position of the adjustment. By adjusting the device though out its entire range and reading the value, you will graph the results on a chart. By graphing a line on the chart, a visual indication is show of the results. By comparing the results displayed on the graph, one can determine the position of the potentiometer. Next a resistor is added. This reduces the range of the potentiometer to half of the previous range.

4.2 Detailed Steps

4.2.1 Monitoring a Potentiometer

☐ Note in the following steps: ADR07H refers to address $009E
ADR07L refers to address $009F
ATD0CTL2 refers to address $0082
ATD0CTL3 refers to address $0083
ATD0CTL4 refers to address $0084
ATD0CTL5 refers to address $0085

☐ Verify power is not applied to CML12S.

☐ Install a potentiometer and 10k resistor on the breadboard per figure.

☐ Connect the following points per figure:

MCU_PORT	Breadboard
GND	GND
+5V	FULL

ANALOG PORT	Breadboard
PIN 15	TAP

☐ Enable JP1, JP2, NO_AUTO and MEM_EN jumpers. Disable MODC and ECS jumper. This will configure CML12S for monitor prompt.

☐ Apply power to the CML12S.

☐ Write $C0 to ATD0CTL2. This sets ADPU bit high which turns analog to digital converter on with fast clear.

☐ Write $00 to ATD0CTL3. This sets A/D converter for eight conversions.

☐ Write $75 to ATD0CTL4. This sets A/D converter for Ten-bit resolution.

☐ Write $90 to ATD0CTL5. This sets A/D converter for multi channel, single conversion.

☐ In the following steps, the A/D results are a 10-bit value but you must read two bytes and combine into a word. The monitor will only read one byte at a time. Whenever "READ WORD" is referenced, write $90 to ATD0CTL5. This starts an analog conversion. Read ADR07H (high byte) and ADR07L (low byte). The result is a word that can be graphed.

☐ Continually READ WORD while adjusting potentiometer for a value as close to $0000 as the adjustment will allow. READ WORD and record value on the graph at 0 turns.

☐ Adjust potentiometer two full turns in the opposite direction. READ WORD and record value on the graph at 2 turns.

☐ Continue to adjust potentiometer two full turns while recording READ WORD value on the graph. Continue for 22 full turns.

☐ Graph result on the graph by drawing a line between points recorded on the graph.

4.2.2 Limiting Range

☐ Connect the following points per figure.

MCU_PORT	Breadboard
GND	GND
+5V	HALF

ANALOG_PORT	Breadboard
PIN 15	TAP

☐ Continually READ WORD while adjusting potentiometer for a value as close to $0000 as the adjustment will allow. READ WORD and record value on the graph at 0 turns.

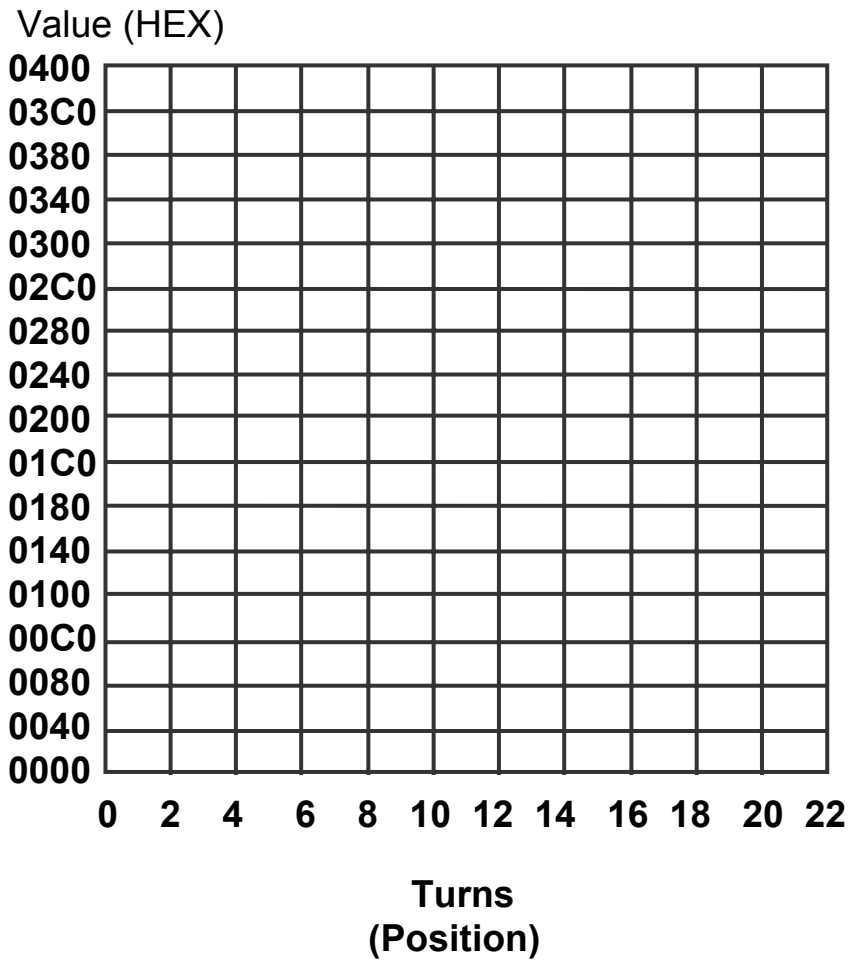
☐ Adjust potentiometer two full turns in the opposite direction. READ WORD and record value on graph at 2 turns.

☐ Continue to adjust potentiometer two turns while recording READ WORD value on the graph. Continue for 22 full turns.

☐ Graph result on the graph by drawing a line between points recorded on the graph.

4.3 Conclusion

As seen in the graph, you can use the analog port to determine the position of the potentiometer. Plus the voltage on the analog port can be calculated by dividing 5 volts by 256 positions and then multiplying by the hex value (converted to decimal). Also by adding a resistor in series with the pot +5 side, the range of adjustment can be reduced by one half. This is helpful in cases where the pot is driven by a higher voltage then the microcontroller input voltage limit. By adding a resistor on the gnd, +5 or both sides of the pot; the range and position of the pot on the graph can be adjusted. The normal analog channel input range is 0 to 5 volts. With a 10k resistor, this can be extended from 0 to 10 volts range. Caution should be excised not to exceed the input range of the analog input. The maximum voltage is the CML12S supply voltage or +5 volts. Pots are used though out the industry. Used for fluid level sensors, position sensors for robots, operator input devices.



5.0 Potentiometer Program

5.1 Program Description

The Pot2A.asm program is used for easily displaying the results of the A/D conversion on the potentiometer position. First the program must setup the A/D channel. The A/D is first turned on, then set for multi channel conversion on channels 0 thru 7. Also set is the single conversion mode. A new line is send to the terminal for displaying new results. A write to ATD0CTL5 starts a new single conversion. A delay is called for conversion to complete (Note: the delay is longer then is required for conversion, it also acts as a delay for displaying lines of data). The X register reads the results and transfers the values to D. Low byte is saved to stack. High byte is send to HEX2, which outputs two hex values. Low byte is removed from stack and send to HEX2. On the terminal you will have four hex values displayed. This is the results of the A/D converting the output of the potentiometer.

5.2 Running Pot2A Program

- ☐ Load program Pot2A.S19 into CML12S. This program is located at $1000, which is internal memory. The source is show below.
- ☐ Call 1000 <enter> on CML12S terminal. This starts the program running and a hex value will start to be displayed.
- ☐ The A/D result value is continually read and displayed. By adjusting the potentiometer, one can get a quick look at the position of the potentiometer.

5.3 Potentiometer Source

```
;
;       Example Potentiometer
;       HC12 code
;       Register Equates
ADR07H:         equ     $009E       ; Analog Result high byte
ADR07L:         equ     $009F       ; Analog Result low byte
ATD0CTL2:       equ     $0082       ; Analog Control
ATD0CTL3:       equ     $0083       ; Analog Control
ATD0CTL4:       equ     $0084       ; Analog Control
ATD0CTL5:       equ     $0085       ; Analog Control
SC0SR1:         equ     $00CC       ; Sci status register
SC0DRL:         equ     $00CF       ; Sci data register
; Setup the A/D converter
                org     $1000               ; program starts here
                movb    #$C0,ATD0CTL2   ; Turn A/D On
                clr     ATD0CTL3        ; Eight Conversions
```

```
                movb    #$75,ATD0CTL4   ; Ten bit Resolution
                movb    #$90,ATD0CTL5    ; Multi Channel, Single Conversion
; Display Results
Loop2
                jsr     CrLf            ; new line
                movb    #$90,ATD0CTL5   ; start a conversion
                jsr     Delay           ; delay for conversion
                ldx     ADR07H          ; read 10 bit conversion
                xgdx                    ; move to Acc D
                pshb                    ; save high byte
                jsr     HEX2            ; send hex
                pula                    ; get low byte
                jsr     HEX2            ; send hex
                bra     Loop2           ; display next digit
; Output A to SCI0
Output:
                brclr   SC0SR1,#$80,Output  ; wait for Xtrans to empty
                staa    SC0DRL    ; send character
                rts
; output right four bits of Acc A as a hex ASCII value
; send to com 1
HEXOUT:
                anda    #$0f        ; mask off bits
                cmpa    #$09        ; compare to number
                bhi     HEX01       ; branch if a thru f
                adda    #$30        ; add standard offset
                jsr     Output
                rts
HEX01:
                adda    #$37     ; change a-f to ascii
                jsr     Output
                rts
; send two hex values in acc A
; send to com 1
HEX2:
                psha                ; save a
                lsra                ; shift right four places
                lsra
                lsra
                lsra
                bsr     HEXOUT   ; send high hex char
                pula                ; restore a
                anda    #$0f        ; only send lower 4 bits
                bsr     HEXOUT      ; send low hex char
                rts
; send car & line feed
CrLf:
                psha            ; save A
                ldaa  #$0a       ; send car
                bsr   Output
```

```
                bsr   Delay
                ldaa  #$0d        ; line feed
                bsr   Output
                bsr   Delay
                pula              ; restore A
                rts
; Delay
Delay:
                ldab    #$10
DelayB
                ldy     #$FFFF
DelayA:
                dey
                bne     DelayA
                decb
                bne     DelayB
                rts
;
```

6.0 Quiz

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

Question Two
What is the maximum voltage allowed on the analog port? A. 5 volts C. 2.5 volts B. No Maximum D. 10 volts

Question Three
What type data is read from the A/D? A. Decimal C. Sine B. Hex D. Matrix

Question Four

What address is ADR07L located?

A. $009E
B. $009F
C. Address $0100 to $01FF
D. External address

Question Five

What is the range with the 10k resistor installed in the 5v side?

A. 0 – 5 V
B. 5 – 10 V
C. 0 – 10 V
D. 0 – 2.5 v

Question Six

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

Question Seven

In this experiment, the A/D is setup for _____ conversion?

A. Digital
B. Single
C. Continuous
D. One shot

Question Eight

How many A/D channels are on the DP256?

A. 8
B. 3
C. 2
D. 16

Question Nine

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

Question Ten

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

Bonus Question
The analog input on the HC12 goes though a?
A. Analog Mux C. Digital Mux
B. Filter D. Decimal conversion

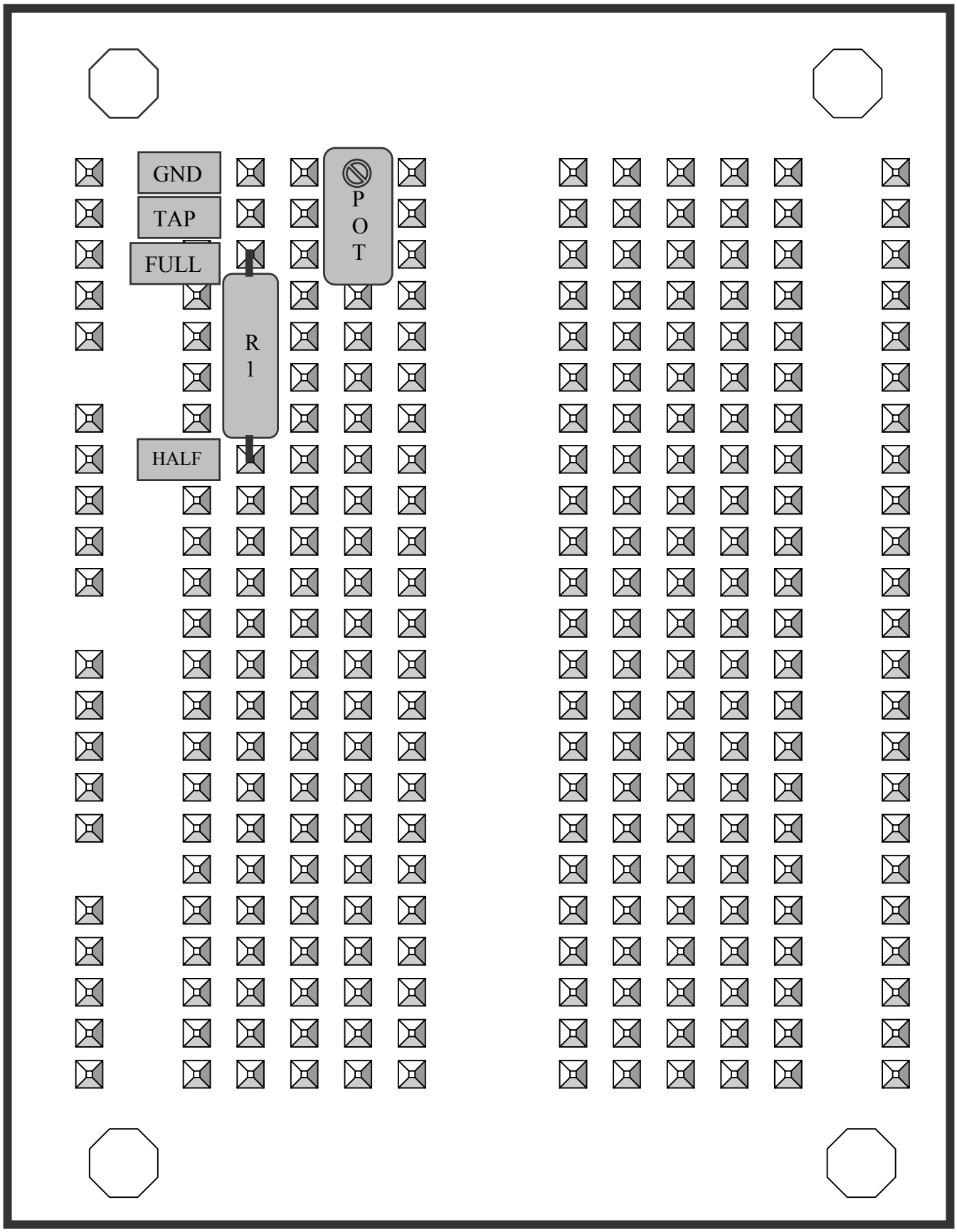
7.0 TROUBLESHOOTING

The development system is fully tested and operational before shipping. If it fails to function properly, consult the troubleshooting section of your user manual.

8.0 Tips and Suggestions

Following are a number of tips, suggestions and answers to common questions that will solve most problems users have with the development system. You can download the latest software from the Support section of our web page at: www.axman.com.

A. Verify jumpers on CML12S are correctly set.
B. Verify lab components are installed correctly.


GND
TAP
FULL
HALF
R
1
P
O
T