

Microprocessor Lab --- Parallel I/O

Using parallel I/O port drive LED

1. Equipment

- M68HC12EVB
- Personal Computer

2. Components

- 1 – Dual digit 14-segment LED, common anode
- 8 – 360 Ω resistors

3. Introduction

The 68HC12 microcontroller contains four I/O ports. The pins used by these ports are shared with other chip functions. These ports, therefore, can not be used indiscriminately. Port A contains three input bits, four output bits, and one bidirectional bit. Port B contains eight output bits. Port C consists of eight bidirectional bits. Ports B and C are not available in expanded mode unless a 68HC24 port replacement unit is used. Port D has six bidirectional bits and shares output pins with the SCI and SPI subsystems of the 68HC12. Port E consists of eight input bits and shares pins with the analog to digital conversion subsystem.

4. Procedure

Connect the circuit as shown in Figure 1 and Figure 2. (Note: you may not need to wire a resistor between each parallel data pin). Write a Program that will read a single digit (0-9) or space from the PC keyboard and display it on the 14 segment LED (a space turns off all of the segments). **See attached code** for getting a character from keyboard. Each bit of port is connected to a particular segment of the display and those segments are lighted up with 0 Volts input (Logic 0). Please also reference attached Parallel IO port address table.

5. Laboratory Write-up

Use the handout distributed on the first lab day as a guide to writing up your lab report. In addition, you must submit a connection diagram and a program. The entire lab report should be typed and submitted during the next laboratory session.

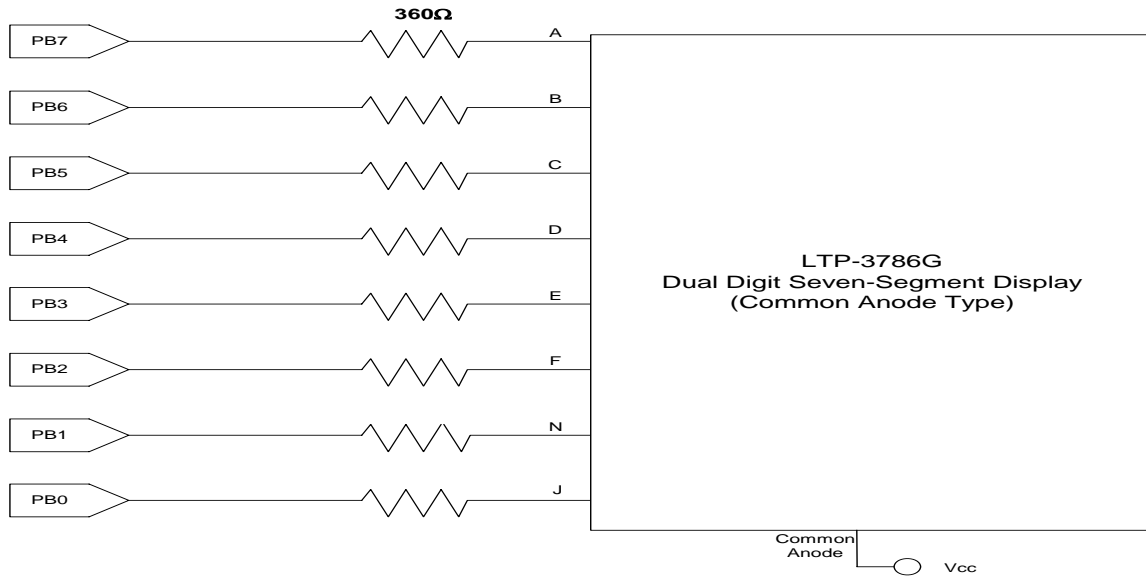
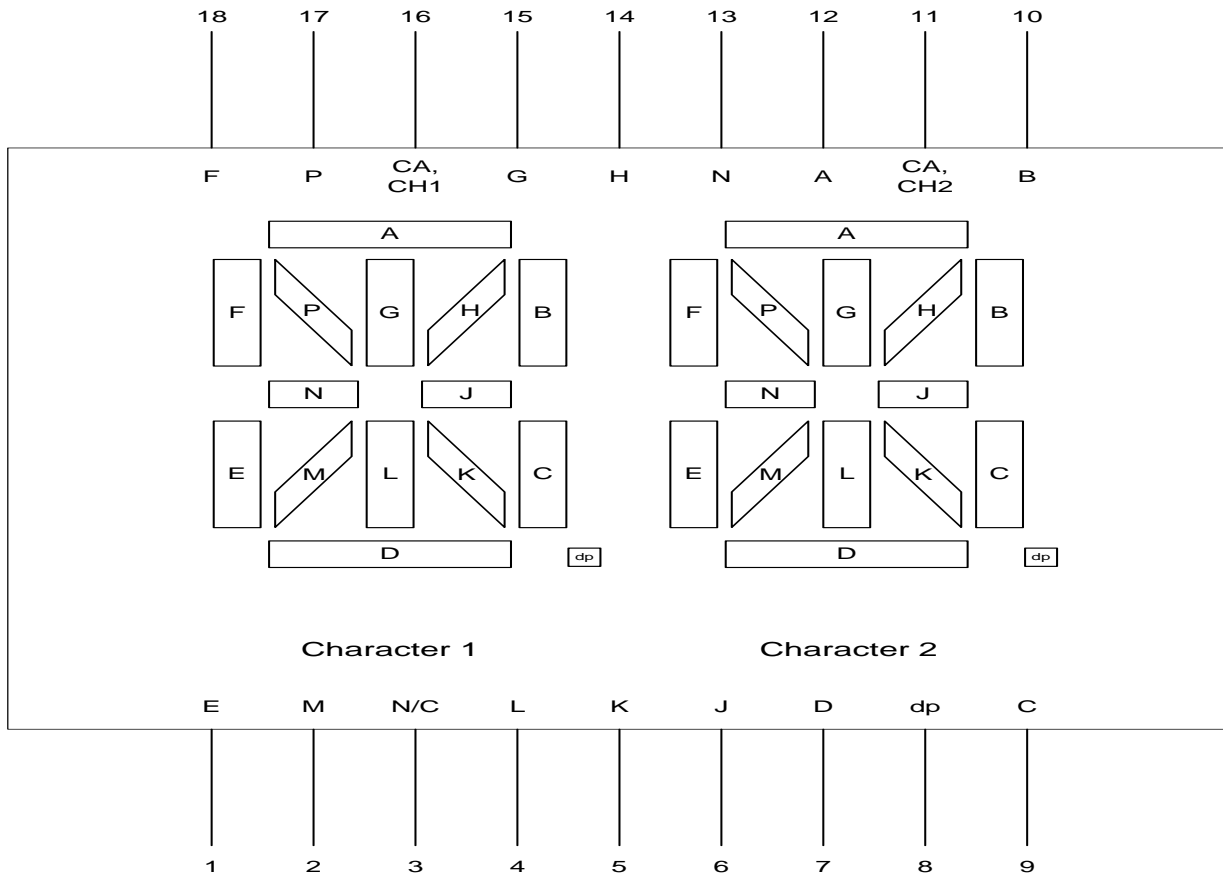


Figure 1. Connection diagram for Lab6.
(Please refer Figure 2. for the pin connection diagram of LTP-3786G)



N/C: No Connection
CA: Common Anode
CH1: Character 1
CH2: Character 2

Figure 2. LTP-3786G pin connection diagram

Attachment:

```
SCOSR      EQU      $00CC
SCODR      EQU      $00cf

getchar    org      $4000
           BRCLR    SCOSR, #$20, * ;wait for RDRF bit to be set
           LDAA     SCODR          ;read char in SCODR into A
           BRCLR    SCOSR, #$80, *
           staa     SCODR
```

Your program should begin from here

```
:
:
cmpa #$20          ;Check if it is "space"
bne getchar
swi
```

HC12 Port Registers

REGBS	EQU \$0000 ; DP256 register bank base address
PTT:	EQU REGBS+\$240 ;portT data register
DDRT:	EQU REGBS+\$242 ;portT direction register
PTS:	EQU REGBS+\$248 ;portS data register
DDRS:	EQU REGBS+\$24A ;portS direction register
PTM:	EQU REGBS+\$250 ;portM data register
DDRM:	EQU REGBS+\$252 ;portM direction register
PTP:	EQU REGBS+\$258 ;portP data register
DDRP:	EQU REGBS+\$25A ;portP direction register
PTH:	EQU REGBS+\$260 ;portH data register
DDRH:	EQU REGBS+\$262 ;portH direction register
PTJ:	EQU REGBS+\$268 ;portP data register
DDRJ:	EQU REGBS+\$26A ;portP direction register