

# CpE 390 Microprocessor Systems

## Lab 5: Parallel I/O

### 1. Introduction

The 68HC12 microcontroller contains a number of parallel I/O ports which can be used to communicate with external peripherals. Each of these ports shares their pins with other chip functions. For example, the pins associated with Ports A and B are also used in “expanded mode” to provide off-chip access to the microcontroller address and data buses. The EVB board operates the microcontroller in “expanded mode”, so we cannot use Ports A and B in our lab sessions.

Port T is a general purpose port that shares its I/O pins with the Timer Module. Providing we are not using the Timer Module, we can use Port T for parallel I/O. There are two registers associated with the operation of Port T:

PTT: EQU \$240 ; Port T Data Register

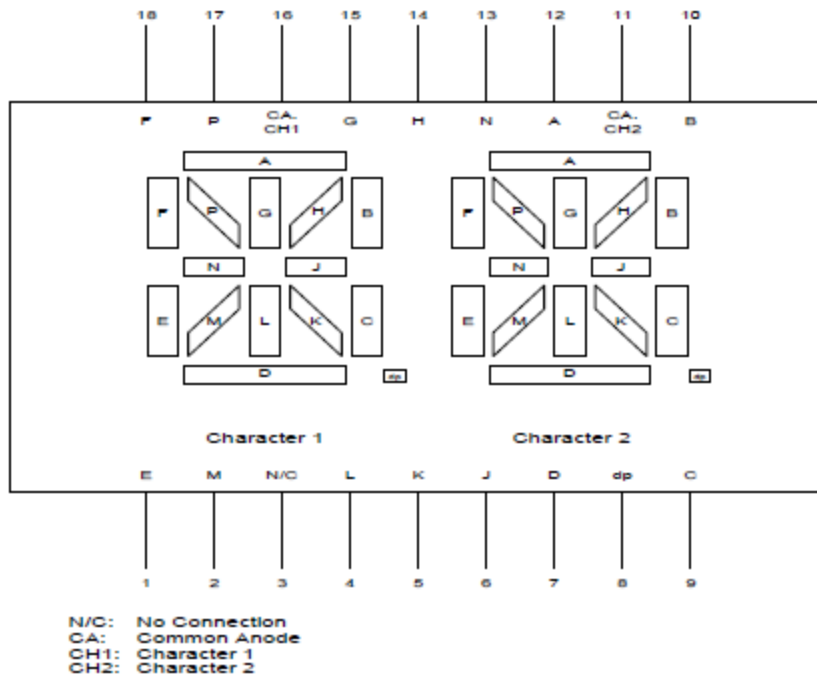
DDRT: EQU \$242 ; Port T Data Direction Register

DDRT defines on a per-pin basis, whether each Port T pin is an input ('0'=input, '1'=output). PTT is the actual data register. Output is performed by writing to PTT. Input is performed by reading PTT.

### 2. Hardware Setup

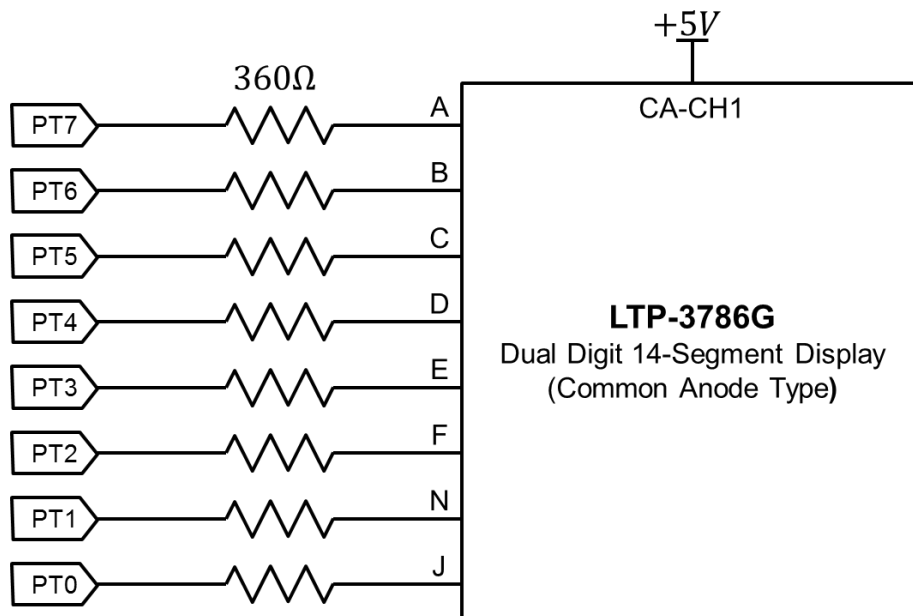
We will be using Port T to write data to an LTP-3786G Dual Fourteen-Segment display. The pin-out for this device is shown in Figure 1. This device can display two characters. We will be using only one of these: *Character 1*. Character 1 consists of 14 light emitting diode (LED) segments plus a decimal point. By lighting the appropriate segments, we can display a variety of alphanumeric characters. In this lab, we will only be displaying the numerical digits 0-9. For this limited set of characters, we only need the eight segments A, B, C, D, E, F, N and J.

Each of the LED segments shares a common anode CA-CH1. If we connect this anode to +5V, we can turn on any one segment by connecting it's cathode to 0V. For example, to turn on segment B, we simply connect CA-CH1 to +5V and pin B to 0V through a current limiting resistor.



**Figure 1. LTP-3786G Pin Connection Diagram**

We will be using the Port T outputs to drive the cathodes of these eight segments. Connect the display to the EVB board as shown in Figure 2. The Port T outputs are current limited. Hence you may not need to use the 360Ω current limiting resistors (check with TA). Note that you must output a '0' from a Port T pin to turn the segment ON.



**Figure 2Connections from EVB to LTP-3786G**

### 3. Software

Write a program that will repetitively read single digits (0-9) from the keyboard and display them on the 14-segment display. Start with the code we used in Lab. 4 to read characters from the serial input (keyboard) and echo them to the serial output (terminal window).

Create a 10 byte lookup table that contains the codes needed to drive the required segments for each digit. For example, the fourth entry in the table would be \$0C. This turns on segments A, B, C, D, N and J, which displays the number '3'. Add this table to the end of your program using a DC.B assembler directive.

Modify your program so that in addition to echoing the character back to the serial output, it also converts the ascii code (assuming it is a numerical digit) to a numerical value between 0 and 9, uses that value to look up the appropriate segment code, and then outputs that segment code to the display.

Assemble, download and run the new program. Check that it operates correctly.

What happens if you type a non-numeric character? Modify your program so that if the character you typed is not between 0-9, you turn off all segments.