

## 7-Segment Board C & Assembly Strategy

This document will suggest basic strategies for creating 'C' and Assembly code for displaying numbers on the 7-segment E-Block.

This E-Block is connected to an upstream board (e.g. the PICmicro multi-programmer board) using 2 connectors - one labelled "ANODE" and one labelled "CATHODE".

The 4 pins used on the "anode" connector are connected to each individual 7-segment display. A '1' on a pin effectively enables the corresponding 7-segment display. The 7-segment displays are labelled as 0 - 3 and this number corresponds to pins 1 - 4 of the anode connector (which in turn would be connected to pins 0 - 3 of a port on the upstream device).

If only one 7-segment display is required, the jumper J4 can be closed to permanently enable one of the displays. This then means that the anode connector does not need to be connected and this will free up a port on the upstream device.

All 8 pins are used on the "cathode" connector and each of these is connected to a particular segment of all 4 7-segment displays. If one of the 7-segment displays is enabled by setting its anode to '1' then a '0' on any of the cathode pins will illuminate the corresponding pins on that 7-segment display.

### Single display example

Thus for a PICmicro microcontroller with the anode connected to Port A and the cathode to Port B, pin A0 on the PICmicro would enable display "0" (i.e. the left-most display), A1 would enable display "1", etc. The pins B0 - B7 would correspond to each individual segment within each 7-segment display, but remember that a logic "0" is required on these cathode pins to turn the LED segment on.

As an example (assuming a PICmicro is used and the anode and cathode are connected to Port A and Port B as described above), to display the number "5" on display 2 you would need to do the following:

- 1) Set A0 - A3 and B0 - B7 as outputs
- 2) Set A2 to '1' and A0, A1 and A3 to '0' - this enables only display 2
- 3) Set Port B to 0x92 - this is the appropriate "code" to display the digit "5"

### Multiple display example

If you want to display different digits on two of the 7-segment displays, then you need to play a trick on your eyes! Here's the strategy to display the number "28" on segments 0 and 1:

- 1) Turn all anodes off by setting A0-3 to '0'
- 2) Set the cathodes (Port B) to 0xA4 - this is digit "2"
- 3) Turn anode 0 on (A0 -> 1)
- 4) Wait for a few instructions
- 5) Turn anode 0 off (A0 -> 0)
- 6) Set cathodes to 0x80 (digit "8")
- 7) Turn anode 1 on (A1 -> 1)
- 8) Wait for a few instructions
- 9) Turn anode 0 off (A0 -> 0)
- 10) Go to step 2

When this procedure is running at a number of MHz, your eyes will not be able to detect the flashing of the LED's and will assume that they are permanently displaying each individual number.

## Converting between a digit and the code for the cathodes

There are a number of ways of doing this. Assembly users could use a lookup-table operation and 'C' users could use a constant array of appropriate codes (where the index for the array is the required digit). Here are the appropriate codes:

```
char CathodeCodes[10] = {0xc0, 0xf9, 0xa4, 0xb0, 0x99, 0x92, 0x82, 0xf8, 0x80, 0x98};
```

Thus the value `CathodeCodes[5]` would be `0x92` which is the appropriate code for the digit "5".

## Further strategies

When displaying different digits on each 7-segment display, you will need to continually set the appropriate cathode and anode values as shown in the example above. This means that you need to be clever about when you perform any calculations or other routines within your code. Chances are you will need to use interrupts and that is beyond the scope of this document.