Application of the Series-80 BCD Interface HP 82941A

Martin Hepperle, 2021

This interface is similar to the GPIO interface HP 82940A, but tailored for input and output of binary coded decimal (BCD) numbers. It can transfer up to 11 digits plus additional control and sign bits. At the time, it was used for interfacing to devices which only had BCD interfaces. For example, many panel voltmeters or A/D converters had parallel BCD interfaces. While these interfaces are very simple and compatible with ASCII number codes, their drawback is that four wires are needed for each digit.

Setup

For demonstration purposes I have set up my interface for output to 2 channels and 5 digits each. As I don't have any devices with BCD interfaces, I built myself two small 7-segment display boxes. Each box is attached to one of the two channels (A) and (B).

To avoid a clash with other interfaces, I also changed the select code to 4 (default was 3).

No	0/1	Function
1	-	n.a.
2	0	
3	0	SC = 4
4	1	

Table 1: DIP switch block S1.

No	0/1	Function	Comment
1	1	Format	Option Format (2 Channels)
2	0	Handshake Lines	Trailing Edge, could be set to "1" to avoid changing register 7
3	0	Data Sense	Negative True
4	0	Signs Sense	Negative True
5	0	Port10 Sense	Negative True
6	0	Control Sense	Negative True
7	0	Flags Sense	Negative True
8	1	Output Enable	Outputs Enabled

Table 2: DIP switch block S2.

Here, the switches 1 and 8 are important for selecting two channels and for enabling output. The remaining switches can be overridden by programming the interfaces control registers.

After booting, the interface is in input mode, even if output has been enabled on DIP switch block S2. In order to prepare the interface for output, the corresponding bits in register 2 have to be set with the following CONTROL statement:

```
CONTROL 4,2 ; 128+64   ! declare channel A and B as output
```

If all outputs shall be deactivated later, either

```
RESET 4                 ! default: A and B as input
```

or

```
CONTROL 4,2 ; 0         ! declare channel A and B as input
```

could be used.

Number of Digits

My setup for the output of 5 BCD digits is using the default settings for 4 mantissa digits and 1 function digit. No exponent and sign digits are used. Thus we see the following values in registers 3 to 5:

Register	Binary	Decimal	Description	
3	0100 0100	68	4 mantissa digits in (A) and (B)	(default)
4	0000 0000	0	0 exponent digits in (A) and (B)	(default)
5	0001 0001	17	1 function digit in (A) and (B)	(default)

For the same result, one could probably also set the number of mantissa digits to 5 and the number of function digits to 0. Note that the total of all digits cannot exceed 11; for symmetry I use only 2 × 5.

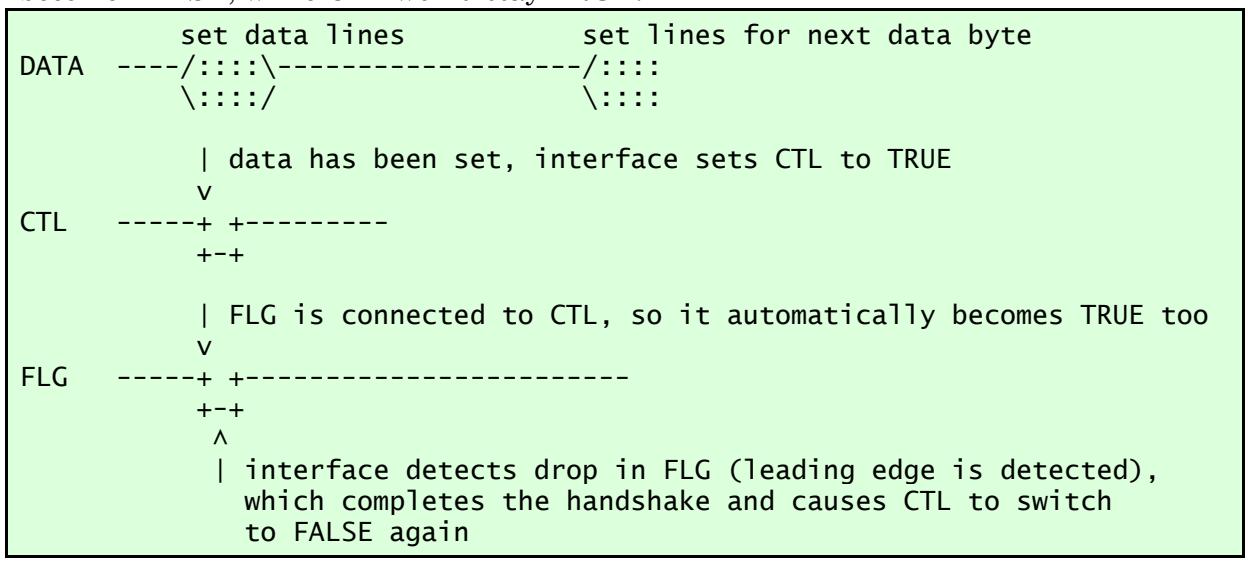
Handshaking

The interface requires proper handshaking for operation. As my output device has no latency, the handshaking lines CTL and FLG on channel A resp. B have been directly connected

For this setup it is important, that the handshake modes of both channels are set to leading edge trigger. This can be established with the following CONTROL statement:

```
CONTROL 4,7 ; 3         ! set handshake modes of (A) and (B) to leading edge trigger
```

If the handshake mode is not set, the interface would hang, waiting for the FLG line to become FALSE, while CTL would stay TRUE.



Note: we use negative logic: negative = TRUE

Output of Numbers

Data for both channels must be output with a single `OUTPUT` statement. Each channel uses 5 of the 11 4-bit BCD ports on the interface:

- Ports 0-4: Channel A, 5 digits, no sign, no exponent, P0: MSD, P4: LSD
- Ports 5-9: Channel B, 5 digits, no sign, no exponent, P5: MSD, P9: LSD

Notes:

Sign bits are not used because my BCD output display shows numeric output only. Therefore, all output must be in the form of exactly 2 x 5 = 10 digits.

Sending any characters other than digits produces an error.

Even if the number of mantissa digits for each channel is set to 4 and the number of function digits is set to 1, only the total length of the output string matters. The interface cannot distinguish between mantissa and function digits.

Example:

- Output "12345" to channel A
- Output "54321" to channel B

As a text string:

```
100 OUTPUT 4 "1234554321"
```

or, using formatted output:

```
100 OUTPUT 4 USING "5Z,5Z" ; 12345,54321
```

or using numeric data or variables and formatted output:

```
100 OUTPUT 4 USING 200 ; 12345,54321
200 IMAGE 5Z,5Z
```

or, if for some reason splitting into more variables is desired:

```
100 OUTPUT 4 USING 200 ; 12,34,5,54321
200 IMAGE 2Z,2Z,Z,5Z
```

In all cases, exactly 10 characters must be transmitted. The first group of 5 goes to channel A, the rest to channel B.

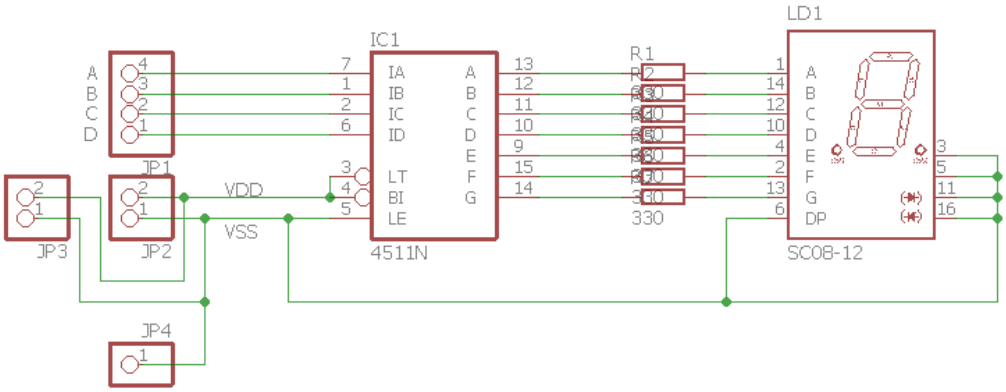


Figure 1: **Simple circuit for output of a single BCD digit to a 7-segment display module. The decimal points are not used. The four BCD wires A, B, C and, D can be soldered directly to JP1 or you install a 4-pin header and crimp a matching header to the 4 wires. JP2 and JP3 allow for chaining the power supply lines of modules and JP4 is for connecting the signal ground from the BCD interface.**

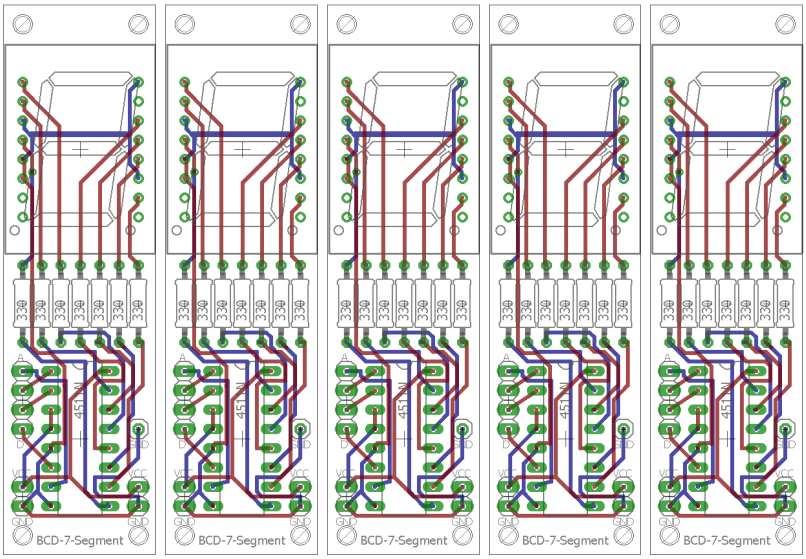


Figure 2: **A set of five printed circuit boards for one output channel.**
I used Ligitek LSD8165-XX-PF 7-segment displays with common anodes, simply because they were available for low cost at the time. Common anode, because, like many interfaces, the BCD interface can sink sufficient current, but not source it. The 330 Ohm resistors limit the current for a supply voltage of 5 Volt, but the units also operate well with 3.6 V from a lithium cell.

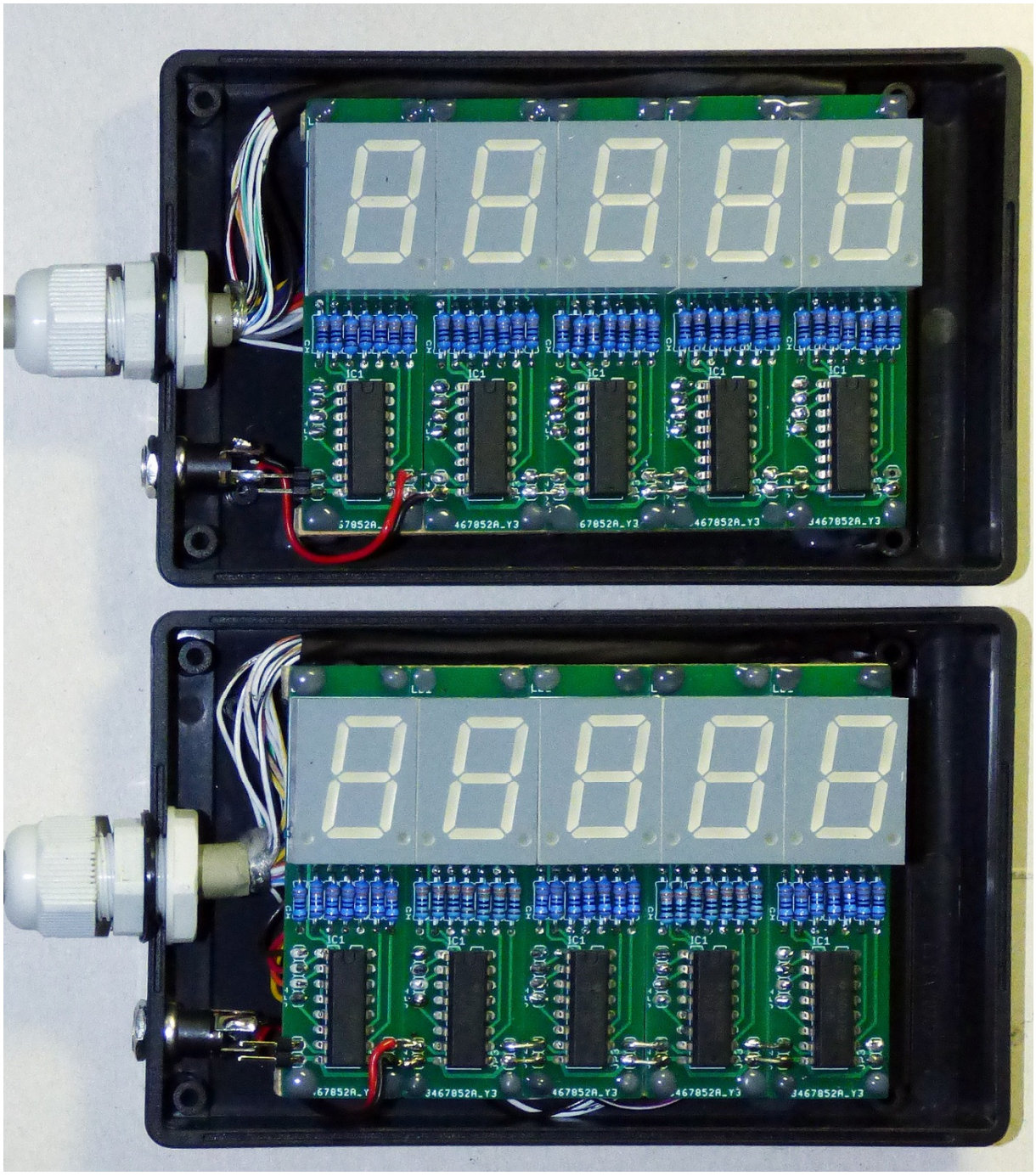


Figure 3: **The two boxes with 20 mm 7-segment displays. Power can be provided by the external 5.5/2.1 mm barrel connector or by connecting an internal battery to the header on the leftmost digit.**