Chapter 14

Serial Interface

We are now at the stage where we are ready to install the I / O (input / output) circuitry to allow us to communicate with the computer via a keyboard and some sort of video display.

14-1. Discussion

There are two ways of communicating with the SK68K:

1. If you already have either a computer terminal (such as a Televideo or Soroq or whatever), or else a computer which can emulate a terminal (using a communications program), then it's easy - we install a serial port on the SK68K by adding two ICs and a connector, and communicate with the SK68K from the terminal via a serial port. This is discussed in this chapter.

2. The other method is to take advantage of the fact that we can use PC-compatible clone components. By installing the keyboard interface and PC-type bus connectors, we can then plug in a clone keyboard and either a monochrome video board or a CGA color board (with, of course, the appropriate monitor). This is discussed in Chapters 15 and 16.

Even if you do not use the serial interface at this time, we will still install it since we have already installed the DUART anyway, and since it is potentially useful to drive a serial printer.

Fig. 14-1 shows the circuitry for the four serial ports that can be installed on the board, although the bottom two ports are optional and seldom used. Since R9, the 3.6864 MHz oscillator at U3, and the MC68681 DUART at U10 have already been installed, at this point we need only add U29, U30, J21, and J22 to complete the top two ports.

As described in the last chapter, the DUART (U10) has a number of functions in the system. In addition to generating the tones for the speaker,

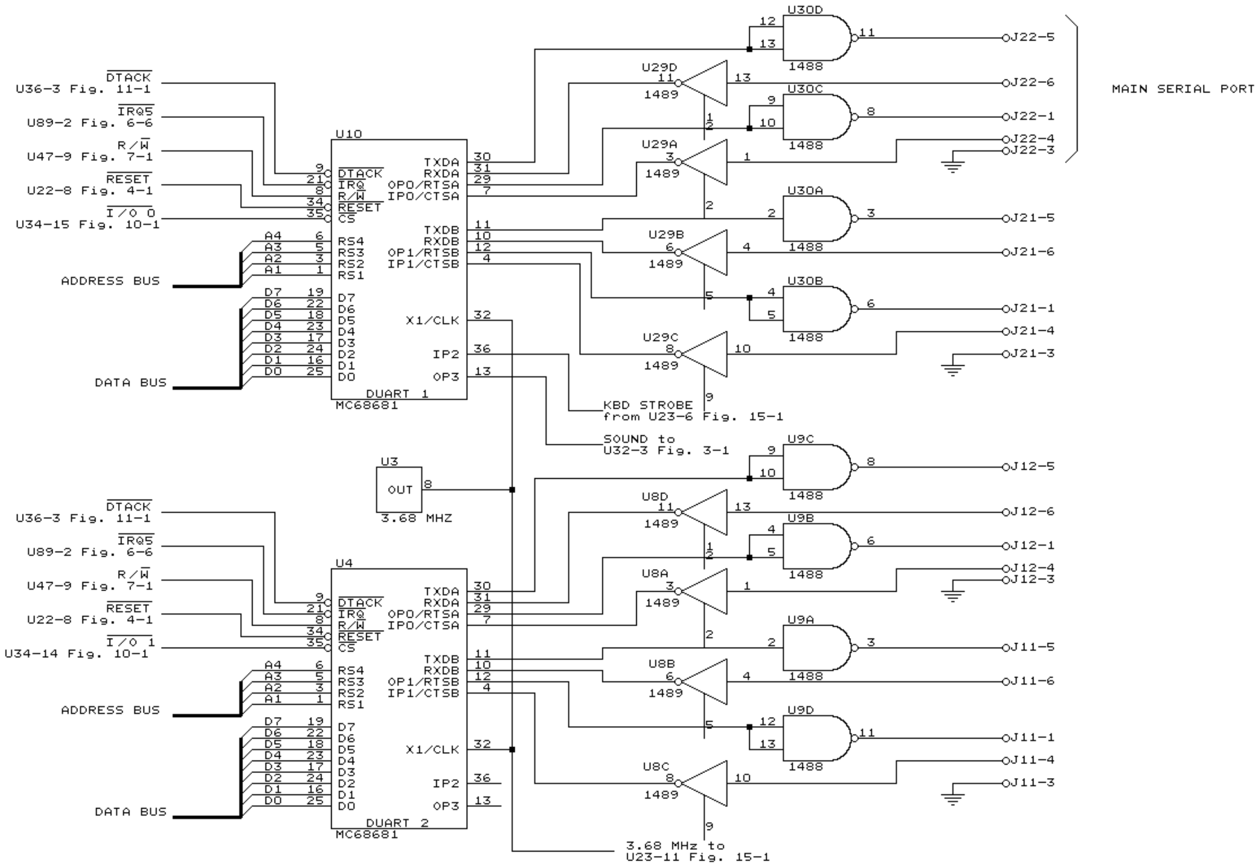

Fig. 14-1. Serial port circuitry.

it interfaces the two serial ports, and even handles the interrupts for many of the system components.

Since the DUART connects to the data and address busses, its internal registers appear in the 68000's memory space as memory locations. It contains two serial ports, called port A and port B. The following addresses are for port A, the primary port used for a terminal:

$FE0007 is the data register. Sending a byte to this address outputs it to the first serial port, via J22; a character input from the port can be read at the same address.

$FE0003 can be read to determine whether the DUART is ready to send or receive a character. The bits at this address are numbered from 0 to 7, with bit 7 on the left and bit 0 on the right. In most cases, bits 0 and 2 are the most important. If bit 0 is a 1, then the DUART has received a character from the serial port, and the character can be read from $FE0007. If bit 2 is a 1, then it is ready to output a character to the serial port, and you should store that character into $FE0007.

Writing to $FE0003 selects the baud rate for the port. The HUMBUG software recognizes the baud rate of your keyboard and automatically sets the DUART baud rate to match (it supports only 300, 600, 1200, 2400, 9600, or 19200 baud), but the baud rate can be changed by placing a different number into address $FE0003. The allowable values are

Baud rate	$FE0003 value
110	$11
300	$44
600	$55
1200	$66
2400	$88
4800	$99
9600	$BB
19200	$CC

Addresses $FE0005 and $FE0015 can cause problems. The Motorola DUART data sheet lists these as "Do Not Access - This address location is used for factory testing of the DUART and should not be read. Reading this location will result in undesired effects and possible incorrect transmission or reception of characters. Register contents may also be changed." Accidentally reading this location may cause your SK68K system to crash.

14-2. Construction

Now mount the following parts on the board:

U30	1488 TTL-to-RS232C converter and its socket
U29	1489 RS232C-to-TTL converter and its socket
J21 and J22	two six-pin header strips

Cut off one center pin and position each header so the cutoff pin is on the side closest to U30; see Fig. 14-2 for details.

The two ICs do the voltage conversion between the DUART, whose inputs and outputs are TTL-compatible (0 is approximately 0 volts, whereas a 1 is about 5 volts), and the serial port wiring (which uses RS-232C levels of about +12 volts for a 0 and about -12 volts for a 1).

As Fig. 14-1 shows, there are four signal leads and one ground lead for each port. On the main port - port A, the one connected to J22 - TXDA is the *transmitter data line*, which sends serial data from the SK68K to a terminal, and RXDA is the *receiver data line*, which receives data from the terminal. RTSA is a *request to send* line which can be used to tell the terminal (or other device connected to the serial port) that the computer is asking for data. Finally, CTSA is a *clear to send* line which can be used by the terminal or other device to tell the computer that it is OK to send data to it. Unless you run special software, regular SK68K software ignores the RTS and CTS lines.

Fig. 14-2 shows how to connect a terminal to the six-pin header. On the terminal end, you will need a DB-25S connector with wiring to pins 2, 3,

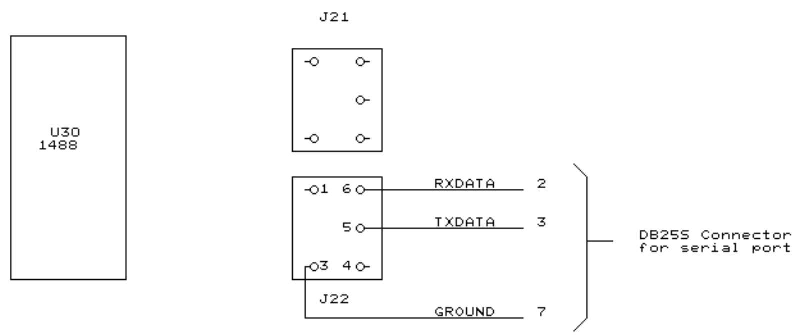

Fig. 14-2. Serial connector wiring.

and 7 as shown. On the computer end, you will need a special connector made by Berg and others (and supplied as part of the SK68K kit). First, crimp (or lightly solder) a Berg 47747 pin on the end of each wire, then insert the three pins into the top end of a Berg plastic shell number 65043-034. Finally, insert a 65307-001 key (a plastic insulating pin) into the hole corresponding to the cutoff pin (labelled with an X in Fig. 4-2) so as to make sure that the connector fits over the header just one way. Don't insert the extra pins into the remaining holes, as they cannot be connected to or removed once installed in the shell.

14-3 Testing

This section assumes that you have a serial terminal, or at least a computer which can emulate a terminal by using a communications program. If you do not have such a device, then skip ahead to Chapter 15.

Connect the DB-25S to the terminal, the Berg connector to J22 (not J21!), and turn on the power. Wait until the beep-boop tone is finished, and then press the RETURN (or ENTER) key on the terminal once or twice. If all goes well, the terminal should display the HUMBUG program signon message and then the prompt.

If the signon message does not appear, it's likely that there is a minor problem with the terminal wiring. First, set the terminal to either 300 or 1200 baud. Then, with a scope or meter (don't use the LED probe from J14-1 or you may damage U32) check the voltage on the RXDATA and TXDATA lines at J22; they should both be negative, between -3 and -15 volts. If the TXDATA line is negative but RXDATA is not, then try swapping the two connections at pins 2 and 3 of the DB-25S connector - both the computer and terminal may be sending data to the same line. If both are negative, check that pressing a key on the terminal makes the voltage on the RXDATA line swing from negative to positive and back; a scope will show this quite clearly, whereas a meter may just show a slight amount of wavering. A lack of this signal means that the terminal is not sending the RETURN code to the computer. Then check the TXDATA line; if it has a signal but the terminal displays nothing, then the terminal is not properly receiving its data. Many terminals require that several pins of their RS-232C connector be properly strapped together (usually pins 5, 6, 8, and 20) before they work.

Apart from possible problems with U29 and U30 (which may not be passing either the received or transmitted signal), there are very few other things which can go wrong at this point since all of the remaining parts of this circuit (including U3 and U10) are already required to make the speaker go beep-boop. (But check U3 anyway - if it is defective, U10 might still be receiving just enough noise that the speaker circuit works yet the serial port does not.)