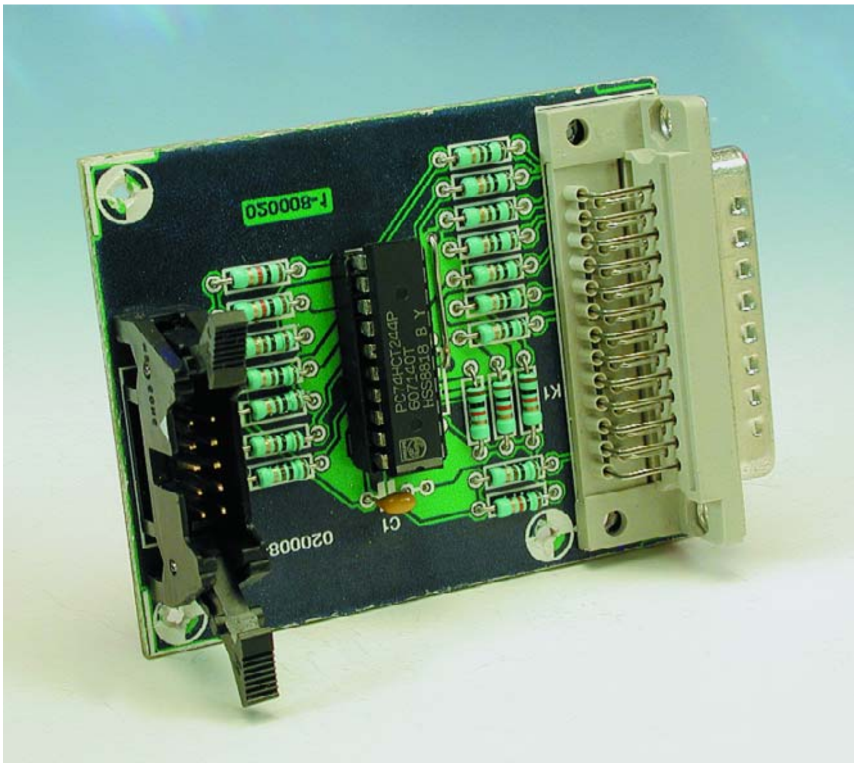
Parallel JTAG Interface

Design by P. Goossens

This circuit is for an interface between the parallel port of a PC and a so-called JTAG interface. This type of circuit was originally developed by a company called Altera, which referred to it as the 'Byteblaster'.



The version described here is compatible with the Byteblaster (electrically it is almost identical). This has the advantage that programs written for use with the Byteblaster can also be used with this interface.

The circuit itself is very simple. K1 is connected to the parallel port of the PC, whereas signals D0, D1 and D6 are brought to the JTAG connector via IC1. The output of the JTAG connector (TDO) is in turn fed back to the PC's BUSY signal via IC1. Pin 7 of the JTAG connector is used to provide an extra output. This signal is brought to the SELECT input of the parallel port via IC1. Some circuits make this signal active while they are in debug mode. This isn't really a requirement, but in some cases this can be useful (assuming that the software supports this feature). This signal could of course also be used to pass different status information back to the PC. IC1 is driven by the AUTOFEED output (active low) of the parallel port. As long as this is high the outputs of IC1 will be in a high-impedance state. This isolates the interface from the JTAG connector, so the target board isn't loaded by the JTAG interface.

The other signals are not specifically required by the JTAG interface. These signals are used by the software to check that the interface has been connected to the (correct) parallel port. Two tests have been made available for this. The first is formed by the direct connection between D5 and the ACK input (active low). The second is made by the connection of D7, via IC1, to the P.END input of the parallel port. This connection only exists when the signal AUTOFEED (active low) is active. These two detection methods are very easy to implement in software. Should a printer be connected to the port, it won't react by printing several pages of gibberish, since the STROBE signal (also active low) has not been used.

Software

Since the interface is designed to test, debug and/or program chips, the software used with it will be dependent on the chip that is connected. But we can still give you a few tips to help you get started quickly with this interface. You should first go to the website of Altera (www.altera.com) and download the software for programming Altera chips via the JTAG interface. You should also be able to find what is called a JAM player. Apart from a Windows version, there is also a Linux version available. The JAM player is in fact an interpreter with its own programming language, which has been specifically designed for use with the JTAG interface. This allows you to control

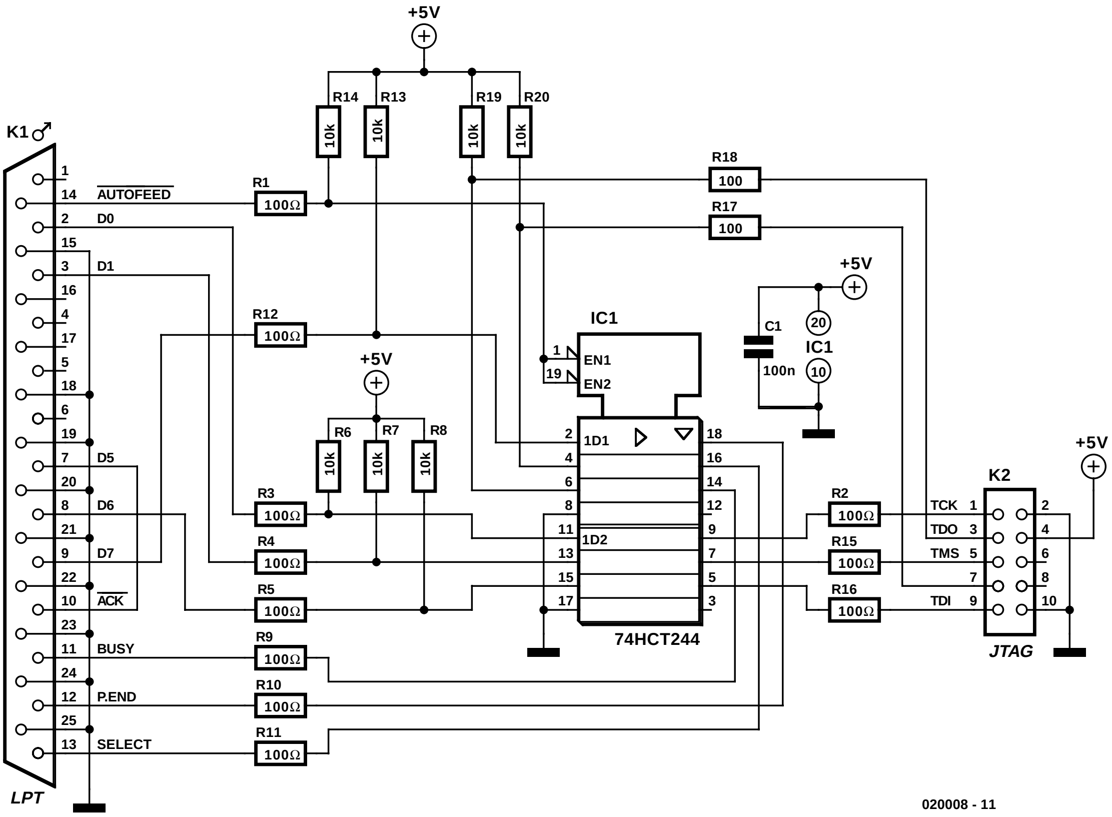


Figure 1. Circuit diagram of the JTAG Interface.

the JTAG interface in a simple manner, using special commands. Apart from the JAM player there are also conversion programs that change several types of file into a JAM format, so that the JAM player can then interpret these files.

There are various projects described on the Internet by people who write software for the JTAG interface. There is little point in giving a summary of the websites here because they are free web pages and regularly move to different servers. Any list will quickly become out of date. In this case a good Internet search engine like Google is the best starting point for finding suitable software.

(020008-1)

COMPONENTS LIST

Resistors:
R1-R5,R9-R12,R15-R18 = 100Ω
R6,R7,R8,R13,R14,R19,R20 = 10kΩ

Capacitors:
C1 = 100nF

Semiconductors:
IC1 = 74HCT244

Miscellaneous:
K1 = 25-way sub-D plug (male), PCB mount
K2 = 10-way boxheader
PCB, order code 020008-1 (see Readers Services pages)

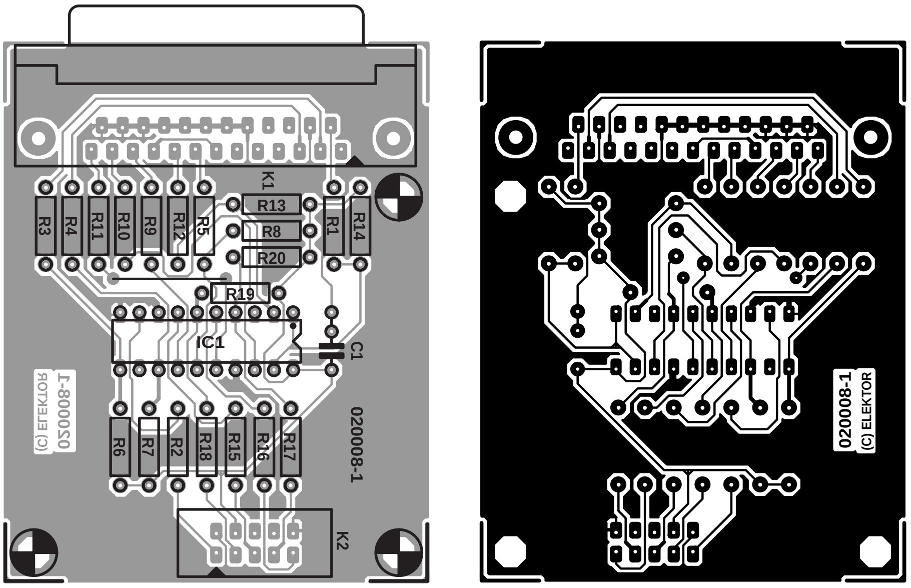


Figure 2. PCB design (board available ready-made).