IET80C31 Development Kit Users Manual

Invector Embedded Technologies

Revision PA1

Östra Ingelstad 1603
273 94 Tomelilla, Sweden

Table of Content

Östra Ingelstad 1603
273 94 Tomelilla, Sweden

Introduction

Dear customer, thank you for purchasing our development kit IET80C31. This is a powerful kit for developing embedded applications and exploring the possibilities of both the microcontroller 80C32 and the Ethernet Controller CS8900A from Cirrus Logic. We have tried our best to provide you with a kit that can be useful both as a platform for your own embedded systems as well as being a valuable tutorial tool that can be used to educate yourself on embedded Ethernet networking.

Kit Content

Before you start working with your new kit please take a moment to check that everything has been supplied. The following should be packed in your kit:

☐ IET80C31 MCU board
☐ IP Dragon (IET8900-5) Ethernet Access Module
☐ Users Manual (This document)
☐ 9 DSUBM <-> 9 DSUBM serial cable
☐ CD-ROM with tools and examples
☐ Plastic bag with 4 legs for the board

What do I need to get started ?

The kit is complete except for having a power source. You need to supply something between 8-15 Volts for the kit to work. The power should be connected to the JP1 power connector. We have mounted a standard screw connector so that the board will be easy to mount in any equipment that you design.

Installing the software

Before starting the board we recommend that you install the supplied software. Insert the CD-ROM into the CD-ROM drive of your PC. The CD should now launch the installation application. Please follow the directions on screen and in the software manual.

Connecting the system to your PC.

Before you apply the power to the IET80C31 board you need to mount the IP Dragon module on the processor board. This is done on the back of the board, just under the serial (JP4) connector. The RJ45 (Ethernet) connector should point the same direction as the serial connector. Make sure the module sits tightly on the connectors and that all pins are correctly aligned.

In order to communicate with the board from your PC you need to connect the supplied RS232 cable from the board to a free serial port on your PC. Connect a power source to your board and start the debugger. The debugger can be found in the newly created folder uC51. Run the file “sld51” and the debugger is started.

Start by selecting the desired serial port and setting the baud rate to 38400 baud.

Östra Ingelstad 1603
273 94 Tomelilla, Sweden

Hardware

Board layout

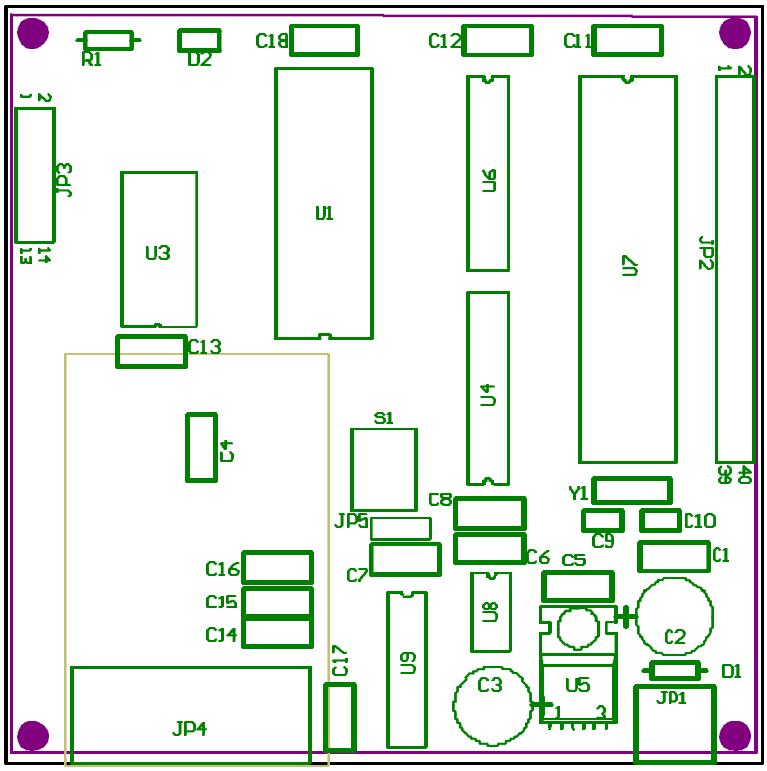

fig. 1 - IET80C31 layout

Fig.1 shows the layout of the IET80C31 application board and below is a short tour of the important parts of the board.

JP1 -> Power Connector.

Use this connector to supply the board with 8 – 15 VDC. The board consumes no more than 100mA so for most purposes a small battery eliminator is sufficient.

JP4 -> Serial Debug Connector.

This is were you connect the IET80C31 to your PC. Use the supplied cable or any other cable with 9DSUBM connectors and pins 2-3 crossed.

JP2 -> Expansion Connector

This connector provides you with all necessary expansion signals such as address bus, data bus and read/write signals. To allow for easy expansion we have provided a RAMDIS signal that allows you to hook up new peripherals in RAM space without reprogramming the on board PLD.

JP3 -> IO Connector

Port 1.0 – 1.7 as well as P3.2 – P3.5 is available on this connector. P3.5 is also connected to the status LED on the board and P3.4 can be selected as a watchdog timer reset signal.

Östra Ingelstad 1603
273 94 Tomelilla, Sweden

JP5 -> Watch Dog trigger selector

The board is equipped with a reset generator that also contains a watchdog timer. On delivery this watchdog timer is strobed by the ALE signal from the MCU and requires therefore no intervention from the MCU. You also have the option to connect this strobe to P3.4 which in turn requires that you have to toggle this port pin at regular intervals. In case of the MCU getting stuck in an endless loop or going astray the watchdog timer will reset the MCU.

U7 -> Standard 80C32 MCU

The board is shipped with a standard Philips 80C32 derivative capable of running at 16MHz. The selected crystal on board is however 14,7456 MHz which is perfect for deriving the high speed (38400 baud) download rate. The Philips device is a standard device that uses 12 crystal clock cycles to derive 1 instruction clock cycle. There are other devices on the market that executes instructions much faster than this. One example is the Dallas Semiconductor DS80C310/320 that requires only 4 clock cycles to execute 1 instruction cycles. This gives you a performance boost of times 3. This higher instruction speed sets higher demands on EPROM and SRAM access times. The IET80C31 is shipped with SRAM and EPROM that meets these demands, thus making it fully possible to replace the standard MCU with a higher speed version.

In our lab we have tested a board with a 22.1184MHz DS80C320 but this required a change of both EPROM and address latch. We had to use a SN74F573 and a 70nS EPROM to make it work. The 100nS SRAM that the board is equipped with works fine, also with a zero state stretch. Although the extra trouble this is 6 times as fast as the standard 80C32.

U1 -> 27C256 EPROM

This EPROM contains the monitor software that allows the SLD51 debugger to talk to the board. You can of course replace this EPROM with your own application software if necessary.

U4 -> Programmable logic device

This device contains the mapping of the different devices on the boards. This is implemented in a reprogrammable device that can be adjusted to fit your own needs. We can supply a custom-made memory map if you do not have the equipment to do this yourself.
The following code have been implemented for this.

```
Name        IET80C31 control device        ;
Partno      iet80c31pld                    ;
Revision    1                              ;
Date        2004-02-12                     ;
Designer    P. Oldberg                     ;
Company     Invector Embedded Technologies ;
Assembly                                   ;
Location                                   ;
Device      f16v8;
Format                                     ;

/** Inputs  **/
Pin 1   =  a6  ;
Pin 2   =  a7  ;
Pin 3   =  a8  ;
Pin 4   =  a9  ;
Pin 5   =  a10  ;
Pin 6   =  a11  ;
Pin 7   =  a12  ;
Pin 8   =  a13  ;
Pin 9   =  a14  ;
Pin 11  =  a15 ;
```


Östra Ingelstad 1603
273 94 Tomelilla, Sweden

```
Pin 12  =  a5 ;
Pin 13  =  a4 ;
Pin 14  =  psen ;
Pin 15  =  rd ;

/** Outputs **/
Pin 16  =  ramoe ;
Pin 17  =  ramcs ;
Pin 18  =  !romcs ;
Pin 19  =  !eamcs ;

/** Declarations and Intermediate Variables  **/

$define  EAMCS   !a15 & !a14 & !a13 & !a12 &  a11 & !a10 & !a9 & !a8 & !a7 & !a6 & !a5
& !a4
$define  ROMCS   !a15 & !a14 & !a13 & !a12 & !a11

/** Logic Equations **/

eamcs = EAMCS;
romcs = ROMCS;
ramcs = EAMCS # ROMCS;
ramoe = psen & rd;
```

A simple function for the device but it gives us the flexibility we need to do changes to the hardware without changing the PCB.

Östra Ingelstad 1603
273 94 Tomelilla, Sweden

IET80C31 Memory map

The physical memory map of the IET80C31 development board is divided into 3 main areas.

1. **0000 – 07FF EPROM (Monitor software)**
2. **0800 – 080F Ethernet Controller (IP Dragon)**
3. **0810 – FFFF SRAM**

These sections are described in closer detail below.

EPROM

The board is equipped with a standard EPROM that contains the monitor software. This software allows the debugger to do high level language debugging on programs downloaded into RAM. The memory select controller is programmed to map the EPROM from $0000 - $07FF (2048 bytes)

Ethernet Controller (IP Dragon)

The Ethernet Controller is found on address 0800 through 080F. In this particular kit the Ethernet module is the IP Dragon which uses the Cirrus Logic part CS8900A. For more information on how to program this device we refer to Cirrus Logic documentation found on http://www.cirrus.com and the drivers that comes with this kit.

SRAM (XRAM)

The rest of the memory space is allocated for your application programs. From 0810 up to FFFF you have RAM to store your programs and data.

On this board PSEN and RD have been AND'ed together which effectively means that the program area and the data (XRAM) area is the same. Internal RAM and SFR's are the same as usual.

Östra Ingelstad 1603
273 94 Tomelilla, Sweden

Software

This kit is supplied with a CD-ROM that contains the development software that you need to start writing your own programs. We recommend that you start by reading the users manual for this software package before actually starting to write you programs.

The package contains an editor, a make utility and a source level debugger that communicates with the board through a serial channel.

Sample Web Server

On the CD-ROM there is also a sample project implementing a TCP/IP stack with a web server using the stack. These examples are all based on the UiP0.9 TCP/IP stack and you should download the latest source code from http://www.dunkels.com/adam/uip before you start on your own project.

The driver (CS8900.c, CS8900.h) contains all the device specific code for using this device with the UiP0.9 stack. It is not suitable for any other TCP/IP stack but can server as an example on how to program the CS8900A device.

You can also download the latest version of the drivers freely from our site were we try to keep them as free from bugs as possible.

Sample TELNET

We have also included a sample TELNET server application. This can be used to demonstrate the ease of remotely configuring and downloading a remotely placed node.

Östra Ingelstad 1603
273 94 Tomelilla, Sweden

Schematics

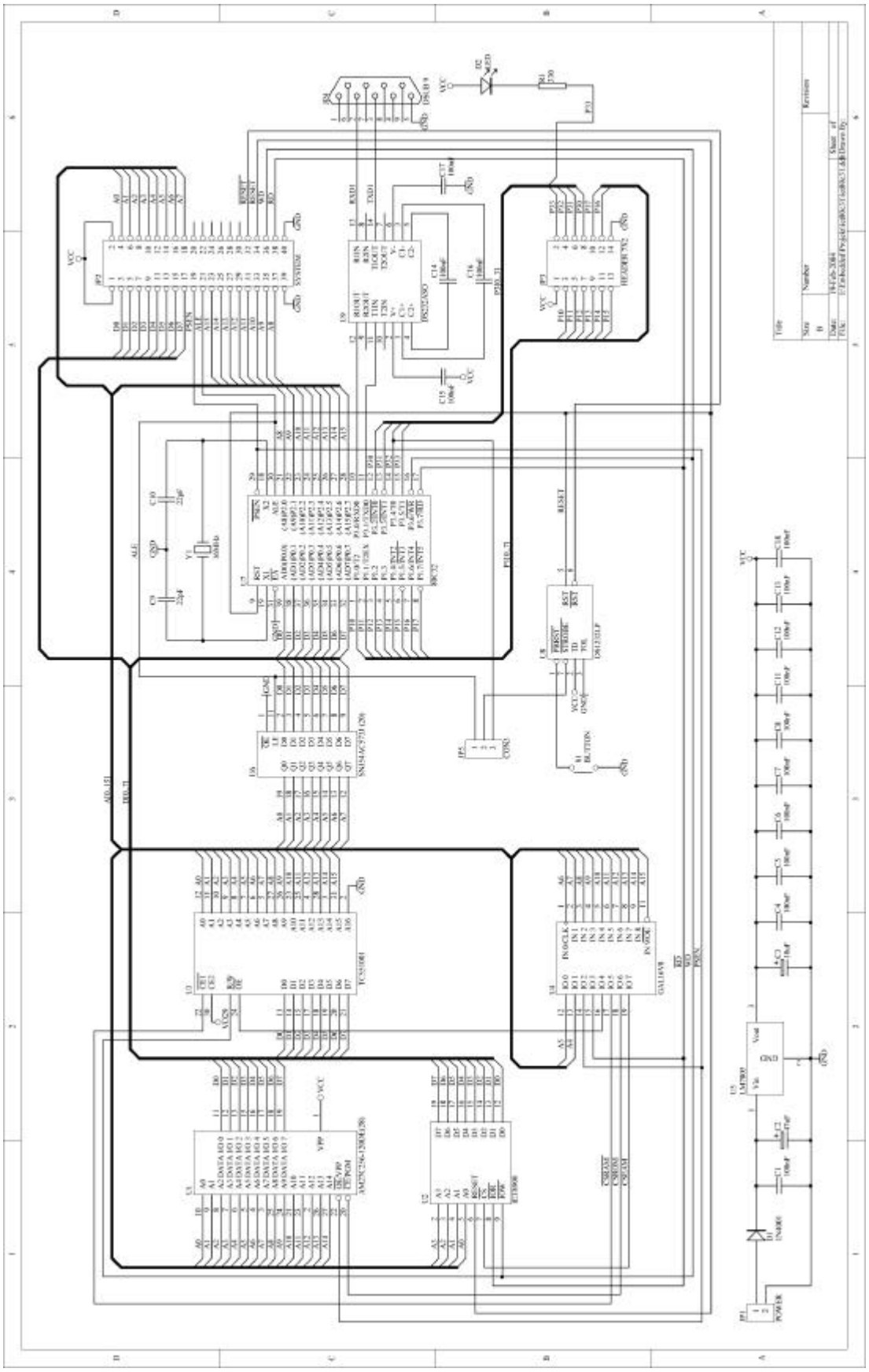

fig 2. Board Schematic

Östra Ingelstad 1603
273 94 Tomelilla, Sweden