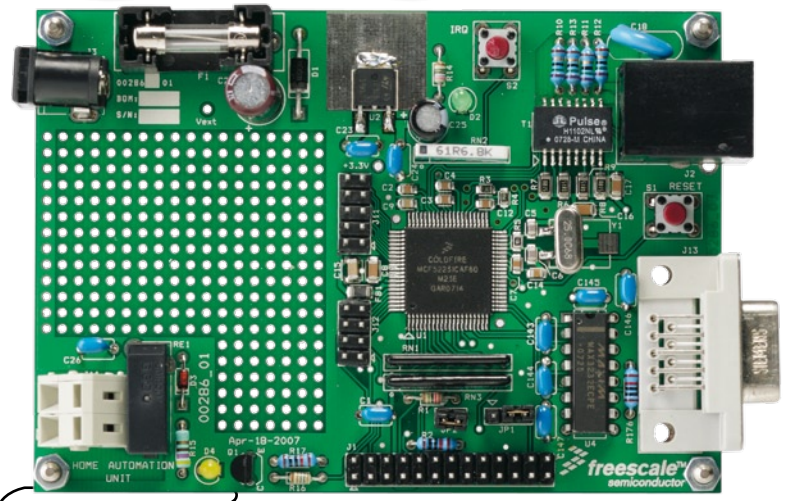
DigiButler



A Coldfire 32-bit home automation server

Part 1: introduction and circuit descriptions

Richard Sumka (Freescale Semiconductor Inc.),
Luc Lemmens & Jan Buiting (Elektor)

This project employs a Freescale Coldfire micro and associated PC software that allows remote switching of electrical loads across networks including the biggest we know – the Internet. The ingredients from the Freescale/Elektor kitchen: 32-bit embedded technology, free software, a low-cost kit for the hardware and free tools to expand the functionality of the server to your own liking.In the first instalment we describe the general structure of the server and the optional Turbo BDM programmer for Coldfire devices.

Out for the night and forgotten to switch off the lights at home, or the heating? This project could be the solution, providing the ability to control equipment remotely over the Internet using a web browser or WAP enabled phone.
Sure, that application alone may look trivial considering the sheer power of the microcontroller used but that's also the crux of the project: it's expandable and totally geared to open-source development as we have made sure that

all resources are available either free (software) or at low cost (hardware).

Networked home automation server

Connecting applications together is fast becoming a necessity rather than an option, especially where Ethernet networking is concerned. This home automation server using a Freescale 32-bit Coldfire device and Freescale software allows remote switching of loads across Ethernet networks and the Internet. And with some ingenuity, simple modifications allow the server to be used for remote sensing and monitoring.

Crossing the internet (and WAP gateways)

Web pages are transferred across the Internet using HyperText Transport Protocol (HTTP). HTTP is a request-response protocol and can be used to send any type of data including binary data. The client – a web browser – requests a web page from a web server and the web server responds with the web page contents. Simple as that may sound, there's a lot of technology behind it all!

As illustrated in **Figure 1**, DigiButler is a mini web server that will happily sit behind an Internet connected router. Alternatively, it may be connected to a local network or directly to a PC. For most of this article we will describe the connection as though it were behind a router.

The unit will accept commands from, and return data to, any Internet-connected PC or WAP telephone that has DigiButler's IP number. Password protection is also provided by the client software.

About the MCF52231

The Freescale Coldfire MCF52231CAF60 in its LQFP80 case is a member of the MCF5223x family of 32-bit connectivity microcontrollers. Its architecture is shown in **Figure 2**. The two key features of the family are the integrated 10/100 Mbit/s Fast Ethernet Controller (FEC) and Ethernet Physical Layer (EPHY); in brief, everything needed to get a single chip application onto an Ethernet network. If you want to delve really deep into this, there's a must-read article available from Eric Gregory [1].

This device also has a CAN 2.0B controller. CAN is commonly used as an

Main specifications

- 32-bit Coldfire MCF52231 microcontroller
- Open-source project
- C source code
- Free CodeWarrior software development platform
- Doubles as a low-cost Coldfire development system
- Connectivity: Ethernet (RJ45), RS232, BDM, analogue, I²C, timers
- 1 relay (on board) suitable for mains loads up to 2 A
- Accessible through Internet or WAP
- Client software with password and username protection
- TCP/IP and RTOS stack with HTTP, DHCP, UDP, ICMP, ARP support
- Easy IP setup
- Ideal for web-driven remote sensing and control
- Kit of parts available from Elektor Shop

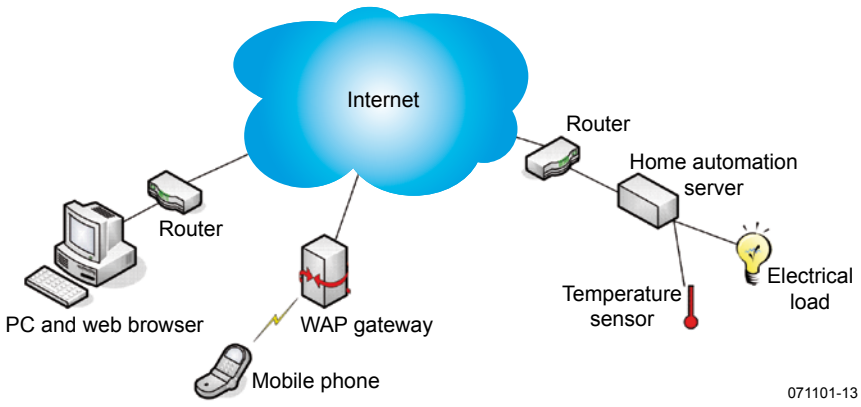


Figure 1. DigiButler gets its commands from any PC connected to the Internet, or a WAP telephone.

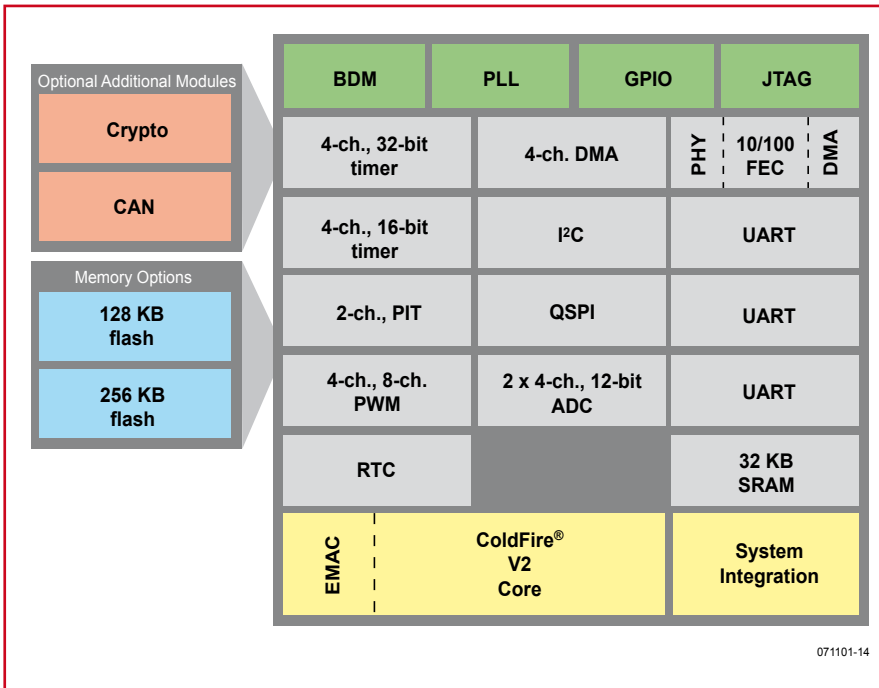


Figure 2. MCF52231 'Coldfire' architecture (courtesy Freescale).

MCF52231 – bits to remember

- 128 kB of embedded Flash memory
- 32 kB of SRAM
- 60 MHz Coldfire V2 32-bit CPU
- Up to 56 bits of general purpose I/O
- Three UARTs
- Serial peripheral interface (QSPI)
- I²C bus interface
- Four 32-bit timer channels with DMA capability
- 4-channel, 16-bit timer for capture, compare and PWM
- 2-channel periodic interrupt timer
- 4-channel, 16-bit or 8-channel, 8-bit PWM generator
- Two 4-channel, 12-bit analogue-to-digital converters
- 4-channel DMA controller
- Up to 73 general-purpose I/Os
- PLL, watchdog, real time clock, range of reset sensors
- On chip background debug module (BDM)
- Single 3.3-volt supply

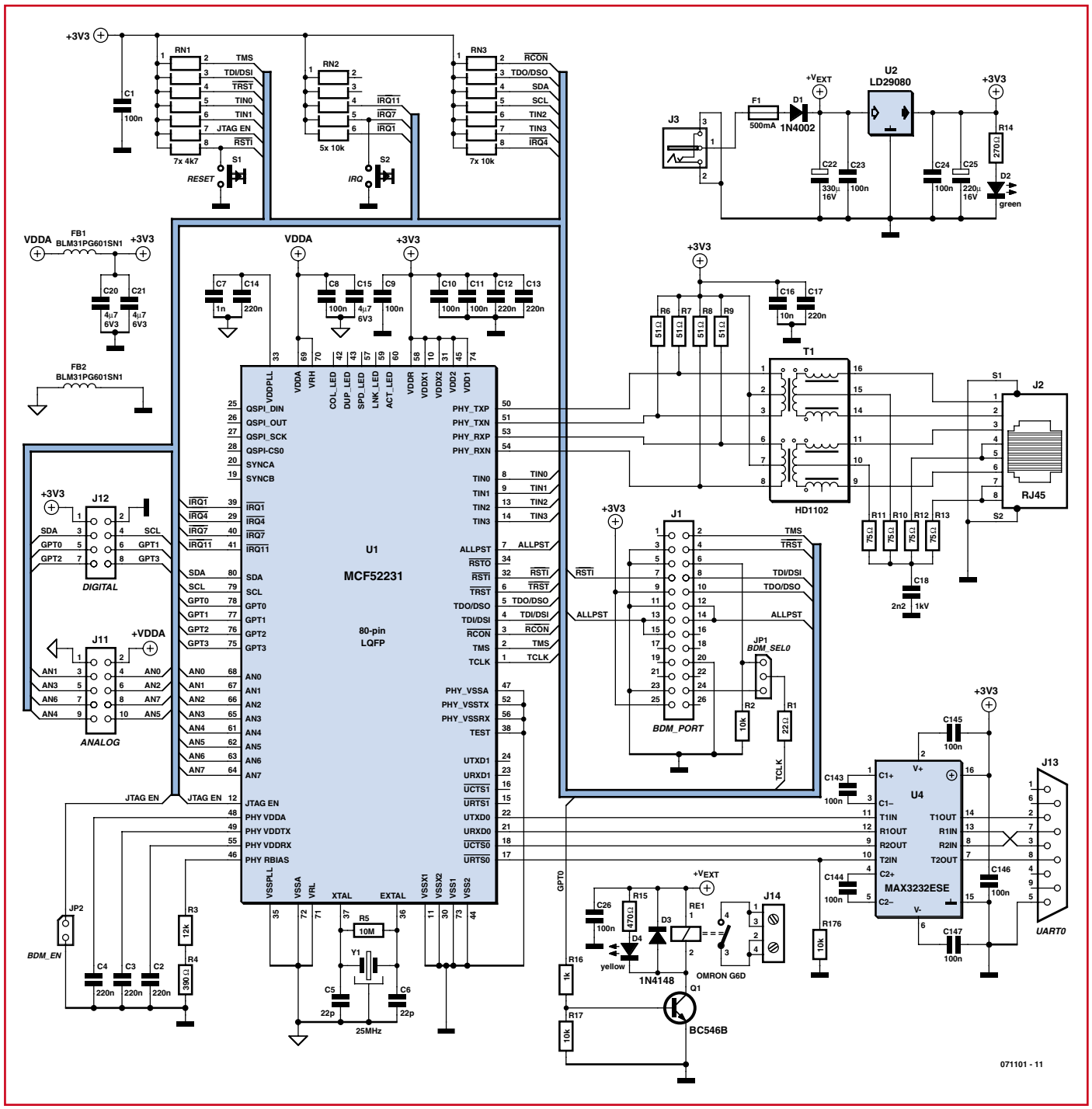


Figure 3. Schematic of the home automation server. The circuit has been designed for expandability — in fact it makes a great development system for Coldfire 32-bit microcontrollers.

industrial control serial data bus because of its suitability for use in real-time communication environments and its reliable operation in conditions of harsh EMI. The MCF52231's bigger brother the MCF52235 also has Cryptographic Acceleration Unit and random number generator for secure hardware encryption. Some of the other important features of the MCF52231 are listed in the **inset**.

Everyone's encouraged to improve & extend the DigiButler C code and let us know the results

Electronics

If we include the transistor and the voltage regulator, there are four active components in the circuit diagram in **Figure 3**. Let's take a tour of the schematic.

At the heart of the circuit sits the Freescale MCF52231 Coldfire device (U1). The 10 or 100 Mbit/s 802.3 ready Ethernet interface is provided by isolation transformer T1 and the physical RJ45 Ethernet connector J2.
Crystal Y1 (25 MHz) sets the clock frequency of the Coldfire microcontroller. This is multiplied up by the device's internal PLL to give a core clock frequency of 60 MHz.
Eight 12-bit analogue inputs are available on connector J11. These are routed directly to the ADC pins of the Coldfire. A further six of the Coldfire's digital input/outputs are available on J12. All can be used as general-purpose I/O and two may be configured to connect to the I²C module in the microprocessor. The two I²C lines, SDA and SCL, are fitted with 10 kΩ pull-up resistors. The I²C can operate at up to 100 kbps with maximum I²C bus loading and timing, and even faster if the bus loading is reduced.
J1 is the BDM (Background Debug Mode) interface, allowing in-circuit debugging of the application code and Coldfire Flash memory erasing and programming. The associated programmer (for optional use) is described further on.
RS232 port J13 is driven by the internal UART of the Coldfire and voltage level translation is provided by U4, the familiar MAX232. A regular RS232 cable should be used to connect the port to a PC, i.e. not a null-modem cable.
Pushbutton S1 is the main Reset and its activation will restart the application code. S2 is directly connected to pin IRQ7 of the Coldfire, with a pull-up to the +3.3 V supply. acts as a general purpose pushbutton input. If you want to 'program it in', feel free to do so!
Jumpers JP1 and JP2 on the board are for programming purposes and will be discussed in part 2.
It's not shown in the circuit diagram, but a large prototyping area on the board gives the user lots of room for experimentation and to expand the board's functionality.
Any low-cost regulated or unregulated power adapter with an output voltage of 5-8V DC at about 500 mA is suitable for powering the circuit. This input minus the drop across D1 is used to supply relay RE1, which has a maximum coil voltage rating of 8 V. As would be expected for such a design, there is reverse polarity (D1) and over-current protection (F1), and an LED (D2) to indicate power on.
Voltage regulator U2 steps the input voltage down to provide the Coldfire device with a stable 3.3 V, which is further decoupled by lots of 100 nF and 220 nF SMD capacitors in key positions. The VDDA supply for U1 is also derived from the +3.3 V line and has additional filtering by ferrite bead FB1 and a pair of low-voltage SMD 4.7 µF capacitors, C20 and C21. Clean as whistle!

Relay control

A key feature of the home automation board is its capacity to control hardware remotely via the Internet. The ability to control mains voltage equipment is especially interesting but requires special precautions. As with any life threatening voltages, safety is paramount and there must be electrical isolation between the low voltage of the board and any mains voltage. Isolation is provided by relay RE1 whose contacts can switch a 250 VAC, 2 A load, the current capacity being limited by the width of the PCB tracks from RE1 to connector J14. Yellow LED D4 shows the relay on/off status.

DigiButler software

The project firmware is a modified version of the Coldfire Lite HTTP server software available free from Freescale and described in Application notes AN3455 [2] and AN3470 [3]. A wealth of information covering the software operation and including training presentations can be found at [4] and [5]. In this project, modifications have been made to the Freescale software to pro-

freescale™ & elektor electronics worldwide

Elektor is again proud and glad to work together with Freescale Semiconductor Inc. for the benefit of its readership. After the great success of the 8-bit MC9S08 SpYder and Accelerometer articles in March and April 2007 we now take a giant leap to a 32-bit embedded system we hope will challenge and inspire the thousands of microcontroller fans among you. The effort is boosted by a kit of parts we're selling at a low price for the DigiButler project.

Historically Freescale has successfully concentrated on the automotive market for its microcontrollers and as a result has held a leadership position there for many years. More recently however, there has been a real push to significantly increase support to the markets served by the distribution network. This increase for the mass market also extends to students and enthusiasts and has led Freescale to work with Elektor.

Interestingly, Freescale has also been asked by some of their largest OEM customers to work with Elektor so that graduates coming into industry are familiar with their products when they start their careers.

Yes, Milord

- The project is open-source with all C code available free for everyone to alter, recompile and flash
- The hardware and software are designed for expansion and experimenting
- You are working with real 32-bit embedded technology
- The project has been designed and tested in close cooperation with Elektor labs
- The PCB in the kit comes with the micro programmed and SMD parts pre-soldered
- The hardware is fun to build on a high-quality board with SMDs pre-stuffed
- There is a large community of knowledgeable Freescale microcontroller users
- The CodeWarrior programming suite is free and easy to use
- East Kilbride is a wet & windy place
- There may well be several Coldfire micros in your new car

vide authentication using username and password and to allow access from WAP enabled phones and web browsers. There are a large number of code modules in the project and all are written in 'C'. As the microcontroller is operating at a whopping 60 MHz there is no need for any assembly language code. In the project settings, 81 kB of Flash memory has been reserved for code space and 45 kB of it for web content. If code optimization Level 1 is used when building the project then only 58 kB of code space is used — give it a try!

The board implements an HTTP web server using a free TCP/IP and real time operating system stack from Freescale. The term 'stack' is used as the software is designed as one protocol stack on top of another, as shown in **Figure 4**. For those interested in the internal workings of the stack, a browse through the project source code will show that the stack supports DHCP, UDP, ICMP and ARP protocols in addition to TCP/IP and HTTP.

Thanks to lots of free information being available on Coldfire TCP/IP stack programming, very little knowledge of the code operation is needed to modify web pages and access the board hardware across the Internet or a WAP phone.

TBLCF: open-source and optional

It's heartening to see that many Freescale micros have been adopted with great enthusiasm by the 'embedded' underground community. Some members have actually developed low-cost alternatives to Freescale's proprietary programming and debugging systems for various microcontroller families, including the high-end ones! In all cases, the concept of BDM is used to access the micros. SpYder [6] is a BDM for MC9S08 micros.

TBLCF is optional, open-source, has USB and costs less than $10 to build

A "Turbo BDM Light Coldfire Interface" (TBLCF) for use with CodeWarrior was developed by Daniel Malik. It is found on the Freescale *68K/Coldfire Processors* forum [7]. In good community spirit Daniel released all relevant material on his design into the freeware domain. If you master the art of 'judicious sampling', TBLCF should not cost more than a tenner for parts.

An important point to mention is that **TBLCF is optional for the present project**. The DigiButler board in the kit supplied by Elektor contains a ready-programmed MCF52231 micro that will not normally require re-programming or debugging. So, TBLCF is for advanced users wishing to modify the DigiButler firmware — everyone is encouraged to do so and show the results.

Daniel Malik's description of TBLCF is exhaustive and eminently presented in free documents and even artwork to make the PCB. There's an associated DLL and a step-by-step software installation guide. Here, we will limit ourselves to a condensed circuit description referring to **Figure 5**, courtesy Daniel.

TBLCF has USB connectivity to the PC. The hardware has two main parts: the MC68HC908JB16 MCU and the BDM interface driver based on a 74VHC14 buffer. The 'VHC14 is used to achieve low-cost translation of BDM signals with voltages anywhere between 3.3 V and 5 V to the 5 V logic of the MCU. The VHC logic accepts overvoltage on inputs, however the output voltage swing is limited by the power rail voltages. When the 74VHC14 is powered by a 3.3 V source, resistors R3 and R4 would not be able to pull the signals above the 3.3 V rail and would only inject current into the power rail of the 74VHC14. Alas, 3.3V is below the minimum High level input voltage of the MC68HC908JB16 and the circuit would not be guaranteed to work. Diodes D2 and D3 have

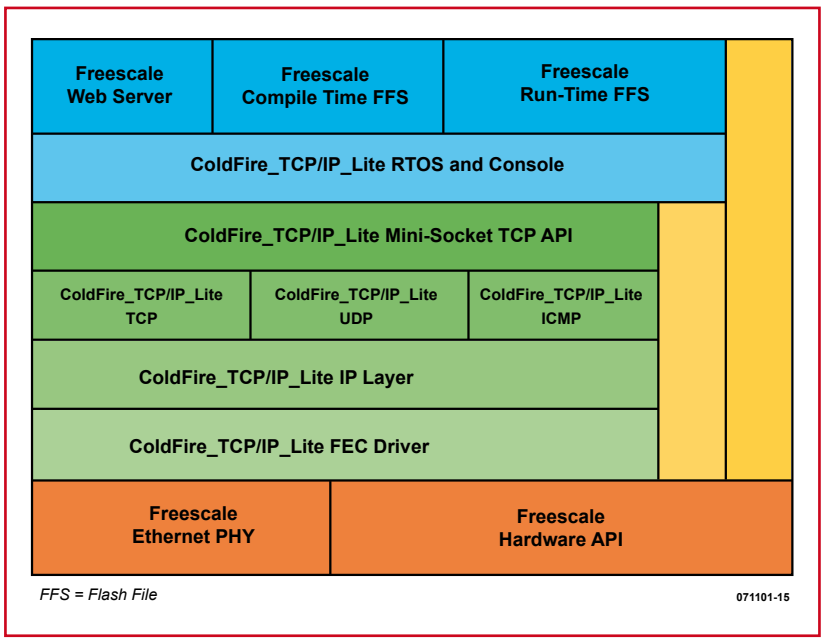


Figure 4. TCP/IP and RTOS stack implemented on the Coldfire micro.

been added to increase the high level voltages. The better alternative, two N-channel MOSFET transistors, would increase the cost and complicate the PCB layout.

The RSTO signal is brought to two different pins of the MCU. This is strictly speaking not needed and a connection to pin PTE1 would be sufficient. However connecting the signal to PTA6 as well simplified the PCB design!

The ColdFire BDM connector has been here for a long time. In the past, boards usually contained a lot of components and were fairly large. A 26-way connector with 0.1" spacing was therefore of a reasonable size. Size of boards is however shrinking and the connector is becoming too large for smaller applications. Two **optional** enhancements have been made to the standard BDM connector:

1. Where the 26-way connector is too large you can use a 10-way subset of the connector (pins 1 through 10). The only signal which is then missing is TA (Transfer Acknowledge) on pin 26, but this is only needed in systems with external memory bus where the debugger is configured incorrectly and accesses an area for which a TA is not generated (neither internally nor externally). So, the probability that it will be needed is quite low and the absence of the signal can be compensated for by a careful use of the debugger.
2. The RSTO signal has been added to pin 1 of the connector, which was so far unused. This enables the interface to detect resets of the microcontroller caused by, for example, the COP/watchdog circuit or a user RESET button.

Note that the above enhancements are suggestions only and the interface will happily operate even with the original 26-way connector. Pins 11 and 12 of the 26-way connector can be removed to make the interface compatible with both the 10-way and 26-way ribbon cables.

Next month

It is planned to have kits for the DigiButler project available with the publication of the May 2008 issue of Elektor. We then finish the article by discussing hardware assembly and test, network connection, Ethernet setup and creating and uploading web pages. For advanced users, CodeWarrior-driven compilation and reflashing of the MPU is also discussed.

(071101-1)

References and Internet Links

Note: documents also available from the project web page:
www.elektor.com/digibutler_en

[1] ColdFire Ethernet, by Eric Gregori.

[2] www.freescale.com/files/microcontrollers/doc/app_note/AN3455.pdf

[3] www.freescale.com/files/microcontrollers/doc/app_note/AN3470.pdf

[4] www.freescale.com/webapp/sps/site/homepage.jsp?nodeId=0162468rH3YTLC

[5] www.emgware.com/

[6] Attack of the SpYder, Elektor Electronics March 2007.

[7] http://forums.freescale.com/freescale/board/message?board.id=CFCOMM&thread.id=624

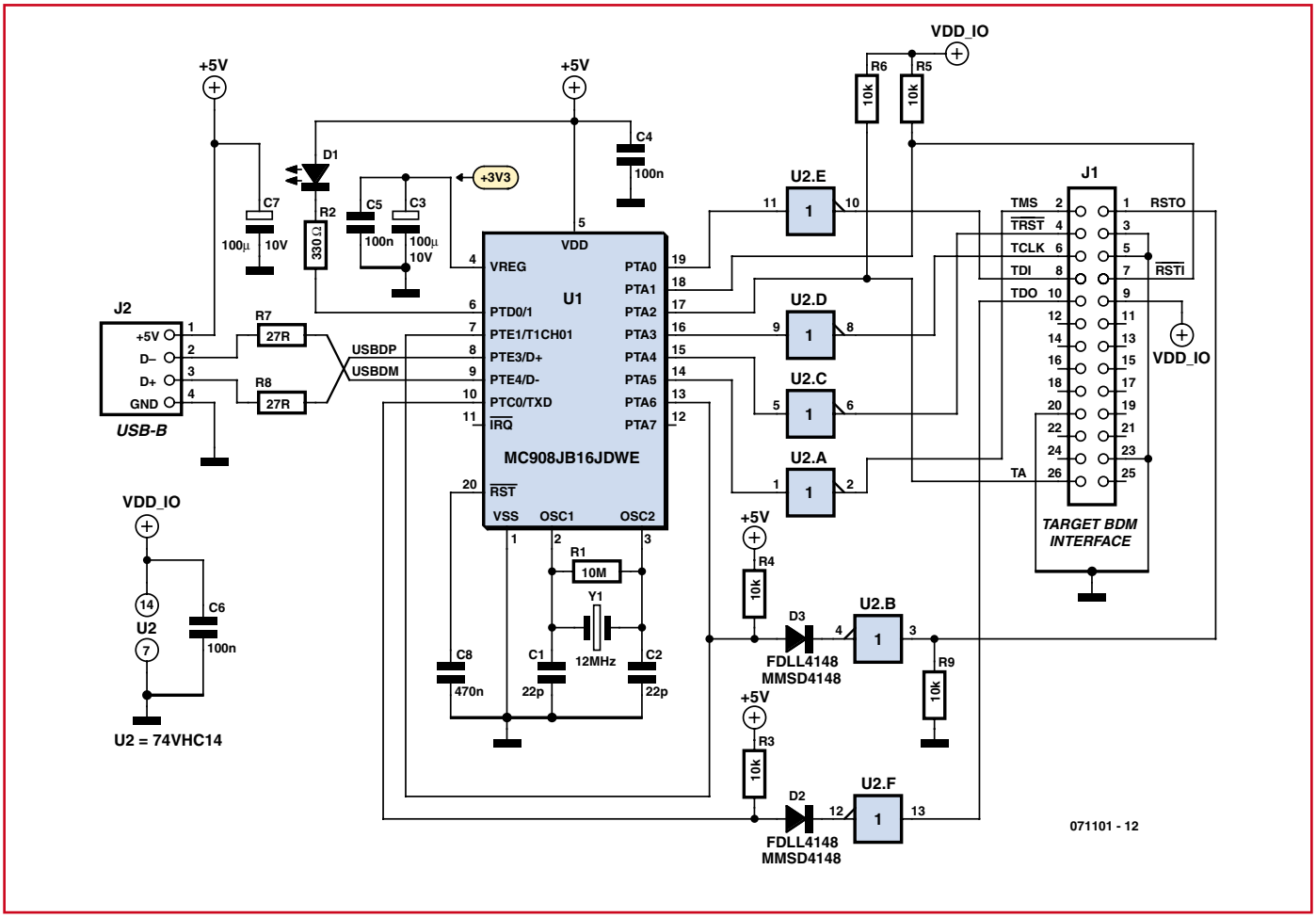


Figure 5. Circuit diagram of TBLCF, the open-source, optional debugger/programmer for Coldfire micros. TBLCF should not cost you more than $10 to build.