assembly, test and software

DigiButler (2)

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

Continuing on from the April 2008 issue, in this the second and final part of the article we go about assembling the kit and connecting up the hardware and software to get DigiButler active on the web and in control of an electrical appliance in your home or office.

Assembly and test

All components used in this project are contained in a kit that's available through the Elektor Shop as # **071102-71** (**Figure 1**). The PCB contained in the kit comes with all but six SMD components ready-soldered and the Coldfire micro ready programmed with the DigiButler firmware. This makes building the project a matter of soldering three dozen or so through-hole components and six relatively large SMD parts including the Ethernet transformer T1.

The component mounting plan of the DigiButler board is shown at true size in **Figure 2**. The associated parts list clearly indicates which parts are prestuffed on the board you get, and which ones have to be soldered.

Assembly of the board should be straightforward using the parts from the kit. Solder the SMD components first and follow with the through-hole components. The SMD parts you have to solder yourself are:

- capacitors C20, C21 (both 4.7μF solid) located at the underside of the board;
- capacitor C15 (4μF7 solid) located between J11 and J12 located at the top side of the board;
- transformer T1 (H1102NL) located at the top side of the board. Triangle marker on the PCB overlay to register with the white dot printed on the device;
- quartz crystal Y1 (25 MHz) located at the top side of the board;
- voltage regulator U2 (LD29080DT33R) located at the top side of the board.

If you use a fine-tipped solder iron and the usual care and precision then mounting these parts should not cause problems. Give the mounting of T1 a thorough visual inspection to make sure no adjacent pins are short-

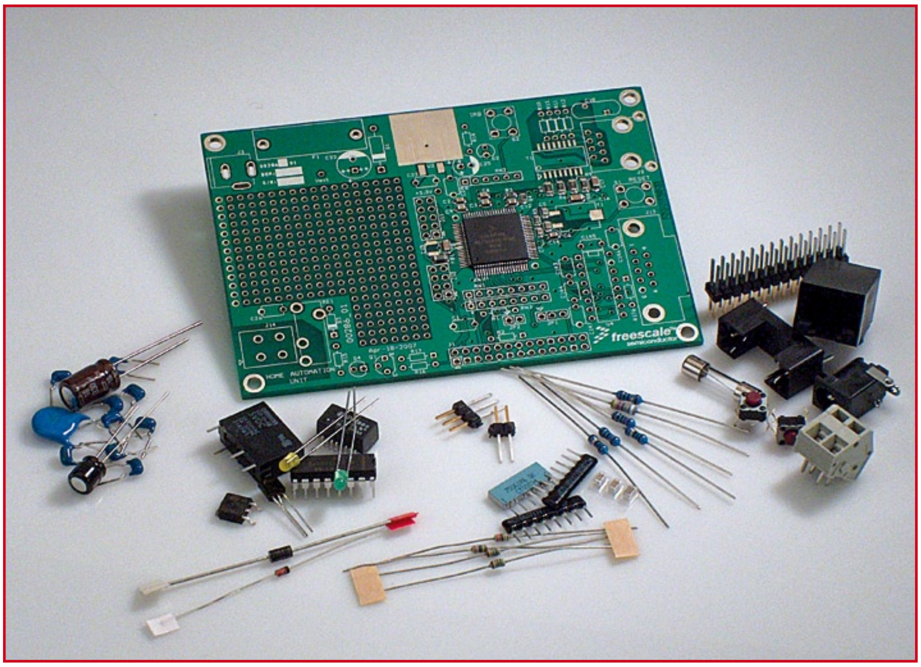


Figure 1. The DigiButler kit ready to be assembled, complete with the SMD-prestuffed PCB. The project is an hour's worth of careful soldering. (packaging not shown; actual parts supplied may differ from photograph)

Exercise extreme caution if using this board to switch power to an AC mains powered load as dangerous voltages will exist on the pins of J14 and RE1. The maximum current switched by the relay contacts is 2 A. If in doubt consult an experienced professional.

circuited by solder blobs.

Now, on with the soldering of the leaded parts from the kit. Voltage regulator U2 has a thermal pad on the underside of the device, the tip of which should be soldered to the nearby copper square on the PCB. The centre pin is not used — the metal pad establishes the ground connection.

The connectors, fuseholder and relay should be assembled last. Pay special attention to J14 and RE1 which should be placed on the board together before being soldered.

The 'pin-1' marker printed on resistor networks RN1, RN2 and RN3 should register with the respective little circles on the PCB overlay. Watch the polarity of radial electrolytics C22 and C25, LEDs D2 and D4, diodes D1 and D3, and finally U4, the MAX3232ECPE.

When placing the assembled PCB directly onto a hard surface, take care to avoid damage to the passive surface mount components at the underside of the PCB. Four 5 mm high PCB pillars secured with M3 nuts will take care of the problem.

The base of RS232 socket J13 may be secured to the board using two M3 screws and bolts. RJ45 socket J2 has two plastic studs that snap secure into the 3 mm holes on the board.

Do not install jumper JP2 (BDM_ENable) and set JP1 (BDM_SEL0) the right-hand position (2-3).

Check the board carefully before applying power. Ensure that all solder joints are good and that there are no shorts. Connect power to the board using either an unregulated or a regulated DC supply with an output voltage of 5-8 V. LED D2 should light up.

I can do it cheaper

Be our guest, although you'll find the price of the DigiButler kit hard to beat. Especially for those among you who insist on sourcing their components locally and doing the microcontroller programming all by themselves we also supply the same SMD-prestuffed PCB with an **empty microcontroller** as Elektor Shop # **071102-1**. 'Empty' means no firmware in the MCF52231, so you will also need the TBLCF hardware and software, or a professional equivalent from Freescale or a third party.

TBLCF hardware

For completeness' sake we recall that the PCB design and construction notes for Daniel Malik's Turbo BDM Light Coldfire Interface (TBLCF) may be found at [7] referenced in Part 1 of this article (April 2008). PCBs for the TBLCF are supplied through the Elektor Shop as # **071102-2**. The bare board is shown in **Figure 3**. You will need TBLCF if you want to make changes to the DigiButler firmware, or use the hardware as a Coldfire development system.

Remember the 'free samples' trick we showed in last year's 3-Axis Accelerometer article? Use the Freescale website again to get a free MC908JB16JDWE chip for the TBLCF programmer. For the project name, enter: 'Elektor DigiButler'. Caution: the supply of free samples is strictly at the discretion of Freescale Semiconductors Inc. Look in Elektor's forum topics on

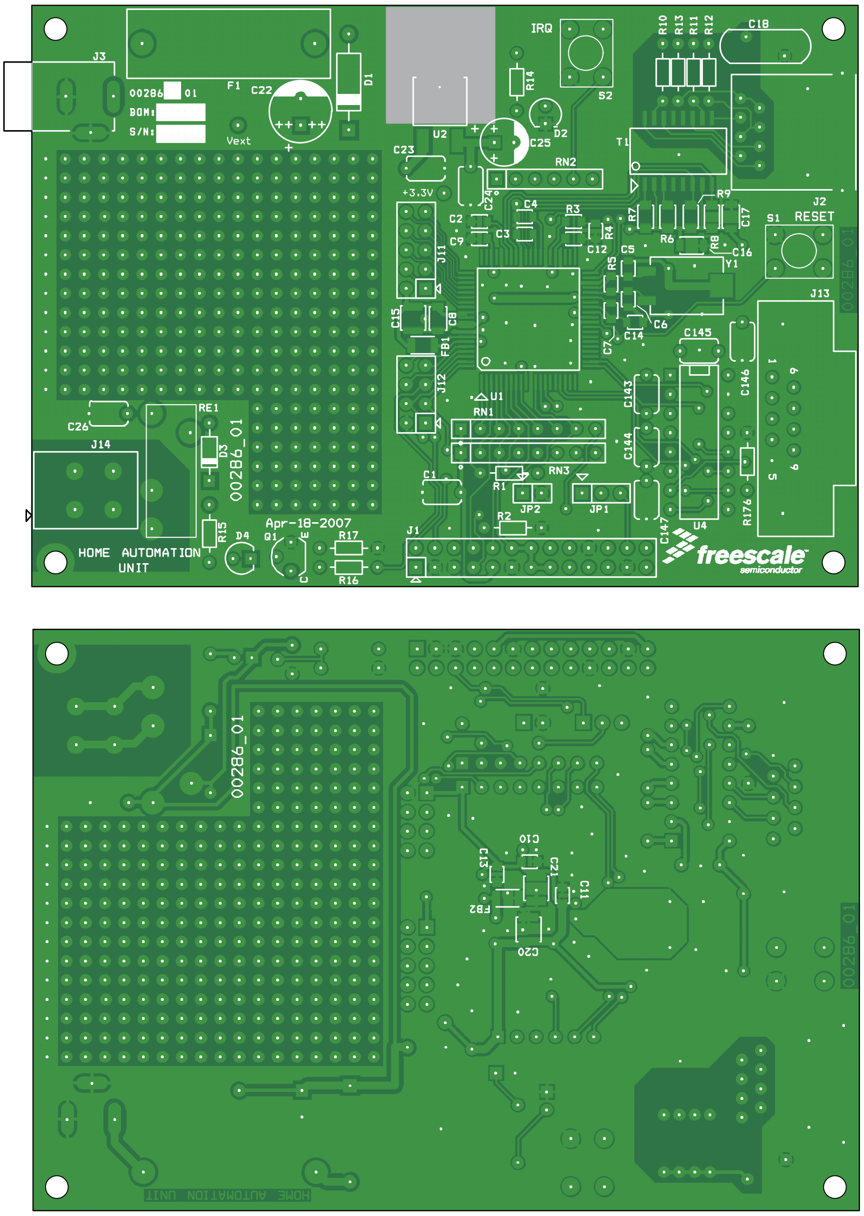


Figure 2. DigiButler board layout. The board comes with all but six SMD parts premounted

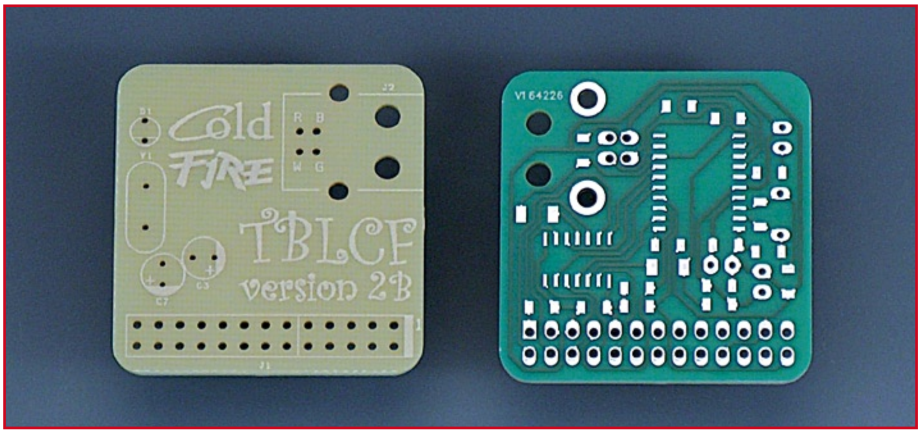


Figure 3. TBLCF board, unstuffed.

the SpYder and Accelerometer projects how others did it.
TBLCF is programmed from the ground up as described in its extensive manual.

Compilation and programming

Code (re)compilation and device (re)programming is only necessary if you

1. have an unprogrammed board # 071102-1 and/or

2. want to make changes to the existing firmware.

The project software including the full C source code is available from the Elektor website as a free download — you'll find it as archive file # **071102-11.zip**. Unpack it into an appropriate directory.

Compilation and (re)flashing the MCF52231 requires **CodeWarrior Development Studio for Coldfire Architectures, v. 6.3**. CodeWarrior is an integrated development environment which includes a project manager, editor, compiler and debugger. Version 6.3 is a free download. At ref. [1] it's marked as "Updates and Patches" but Elektor has 'inside information' that it's actually a full install version! The download is about 140 MB. Type 'Elektor DigiButler' when you are asked for the project name.
At the time of writing, the 'Special Edition: CodeWarrior for ColdFire Architectures' (v. 7.0) revealed some issues with the MCF52231 micro we're using. An update/patch or service pack for 7.0 has been announced by Freescale. It should be well worth the wait considering that the suite can compile up to 128 k of code in 'C' and unlimited assembly code. The Special Edition can be downloaded at [1] too — sit back & relax for 270 MB or so.
Programming and debugging also needs a background debug pod such as TBLCF or the USB-ML-CFE [2] from P&E Micro. A detailed procedure to do the firmware flashing is given in the **'(re-)Flashing your board'** inset.

Network connection

The most important part of the project is connecting DigiButler to a network, and for this we need an IP address for the board. An IP address is similar to a telephone number and allows the board to be identified by other devices on the network. By default, the board software is configured to request an IP address from a DHCP server, although it can be reconfigured to have a fixed, or static, IP address. Most home cable or DSL modem routers have built in DHCP server functionality and are able to provide local IP addresses.
To link to your network, connect the DigiButler board to your router using a standard RJ45 cable. Next, connect the

COMPONENTS LIST

1. Components pre-fitted on board

Resistors
R3 = 12kΩ, SMD 0805
R4 = 390Ω, SMD 0805
R5 = 10MΩ, SMD 0805
R6,R7,R8,R9 = 51Ω, SMD 1206

Capacitors
C2,C3,C4,C12,C13,C14 = 220nF ceramic, SMD 0805
C9,C10,C11 = 100nF ceramic, SMD 0805
C17 = 220nF ceramic, SMD 1206
C5,C6 = 22pF ceramic, SMD 0805
C7 = 1nF ceramic, SMD 0805
C8 = 100nF ceramic, SMD 1206
C16 = 10nF ceramic, SMD 1206

Inductors
FB1,FB2 = BLM31PG601SN1 ferrite bead, SMD 1206

Semiconductors
U1 = MCF52231CAF6 (Freescale)*

2. Components to fit on board

Resistors
R1 = 22Ω SFR16S
R2,R17,R176 = 10kΩ SFR16s
R15 = 470Ω
R10-R13 = 75Ω SFR16S
R14 = 270Ω
R16 = 1kΩ SFR16S
RN1 = 7-way 4kΩ7 SIL
RN2 = 5-way 10kΩ SIL
RN3 = 7-way 10kΩ SIL

Capacitors
C1,C23,C24, C26,C143,C144-C147 = 100nF ceramic, lead pitch 5mm
C15,C20,C21 = 4µF7, solid, SMD1206
C18 = 2nF2 1000V ceramic (TDK)
C22 = 330µF 16V, electrolytic, radial
C25 = 220µF 6.3V, electrolytic, radial

Semiconductors
D1 = 1N4004
D2 = LED, 3mm, green
D3 = 1N4148
D4 = LED, yellow, 3mm
Q1 = BC546B
U2 = LD29080DT33R (STMicro)
U4 = MAX3232ECPE

Miscellaneous
F1 = 0.5A slow, 5x20mm
J1 = 26-way double-row pinheader
J2 = RJ45 socket, PCB mount
J3 = DC adaptor socket, PCB mount
J11 = 10-way double-row pinheader
J12 = 8-way double-row pinheader
J13 = 9-way sub-D socket (female), angled pins, PCB mount
J14 = 2-way terminal block, 5mm lead pitch
JP1 = 3-way pinheader with jumper
JP2 = 2-way pinheader with jumper
RE1 = G6D-1A-ASI-DC5 (Omron)
S1,S2 = pushbutton, EVQ-PAE05R Panasonic)
T1 = H1102 Ethernet transformer (Pulse)
Y1 = 25MHz quartz crystal
Fuse holder, PCB mount
Kit of parts, Elektor Shop # **071102-71**, contains all parts and SMD-prestuffed circuit board (see text)
PCB, Elektor Shop # **071102,** available separately. Marking: 00286_01 "HOME AUTOMATION UNIT"
PCB for TBCLF, Elektor Shop # **071102-2 ****

* ready-programmed on PCB in kit 071102-71; blank on PCB 071102-1.

** optional, see text.

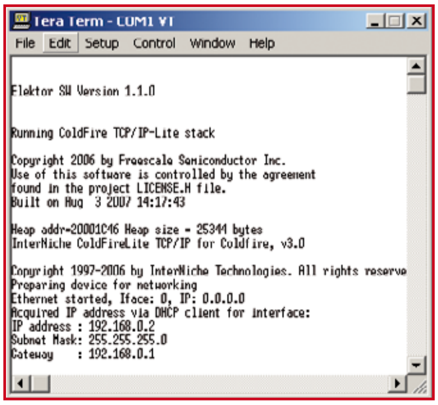
board to a PC's COM port using a regular RS232 cable. USB-to-RS232 adapters are often problematic so an (older) machine with a real serial port (COM) is preferred. If necessary, borrow the kiddies PC.
Start up a terminal session such as Minicom for Linux or HyperTerminal or TeraTerm for Microsoft Windows.
The communication settings are

- **115,200 databits/second**
- **8 data bits**
- **no parity**
- **1 stop bit**
- **no flow control**

Power up the board and push switch S1 to reset it. This will cause the application software to communicate with the DHCP server in the router to acquire an IP address for the board. On the PC screen, the terminal window will display the board IP address, like **F 4,** indicating connection to the network and ability to be accessed remotely. Note that it takes a few seconds for the board to obtain an IP and include it in the message sent to the terminal. Got the complete message? Congratulations, your hardware is working.
If you just want to connect the board direct to a PC, you will need a **crossover RJ45 cable** and a **fixed IP address** in the application software.

Ethernet setup and access security

We have assumed so far that DigiButler will be connected to the Internet via an Ethernet router. A router allows many computers on a local network to share the same internet connection. It also prevents unwanted Internet visitors accessing those same computers unless authorised.
Preventing access to your private DigiButler could be a problem if it were not for a router feature called Port Redirection or Port Forwarding, see **Figure 5** for a typical example.Port Forwarding lets through Internet traffic destined for a given IP address and port and then sends it to a computer or other device on your local network. Let's say the local IP address of the server is 192.168.0.2 and the address of your router – given to you by your Internet Service Provider (ISP) — is 86.131.222.120. Login to the administration pages of the router and change the router settings to forward any incoming traffic destined for port 80 to DigiButler at address 192.168.0.2. Anyone who now enters http://86.131.222.120 in their web browser will be taken to your DigiButler board.
Be aware that you probably won't be able to see the server if you try this from a PC on the same local network. Instead you will have to use http://192.168.0.2.
Another important point to note is that each port can only be forwarded once. It's not possible to have port 80 forwarded to multiple IP addresses so you won't be able to run a web server on a local network and also run DigiButler as they both use port 80.



```
Elektor SW Version 1.1.0


Running ColdFire TCP/IP-Lite stack

Copyright 2006 by Freescale Semiconductor Inc.
Use of this software is controlled by the agreement
found in the project LICENSE.H file.
Built on Aug  3 2007 14:17:43

Heap addr=20001C46 Heap size = 25344 bytes
InterNiche ColdFireLite TCP/IP for Coldfire, v3.0

Copyright 1997-2006 by InterNiche Technologies. All rights reserve
Preparing device for networking
Ethernet started, Iface: 0, IP: 0.0.0.0
Acquired IP address via DHCP client for interface:
IP address : 192.168.0.2
Subnet Mask: 255.255.255.0
Gateway    : 192.168.0.1
```

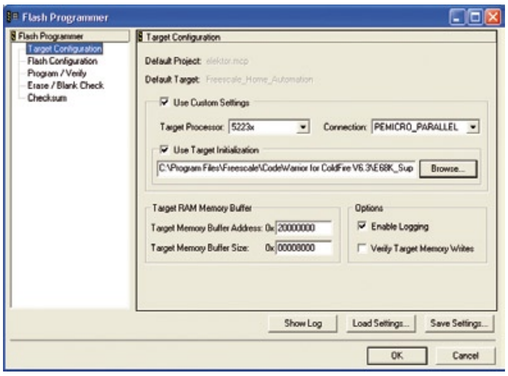
Figure 4. **"It's alive Jim!"**
Output from serial port after DigiButler is reset.

The end result: network, web and WAP

The home automation server is ready for access over a network as soon as the Coldfire device has been programmed and the network cable is

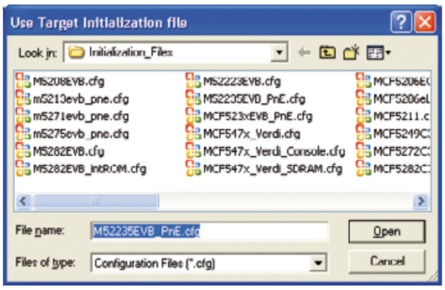
(re-)Flashing your board

The following procedure was recorded with a PE Micro parallel programmer / debugger pod connected to J1 on the DigiButler board, a laptop PC running Windows XP and a downloaded version of CodeWarrior 6.3. TBCLF may require different jumper settings and steps, please consult the TBCLF documentation. Updates to this procedure may become available at www.elektor.com/digibutler_en.



Target Configuration
Default Project: elektor.mcp
Default Target: Freescale_Home_Automation
☑ Use Custom Settings
Target Processor: 5223x
Connection: PEMICRO_PARALLEL
☑ Use Target Initialization
C:\Program Files\Freescale\CodeWarrior for ColdFire V6.3\E68K_Sup
Target RAM Memory Buffer
Target Memory Buffer Address: 0x 20000000
Target Memory Buffer Size: 0x 00008000
Options
☑ Enable Logging
☐ Verify Target Memory Writes

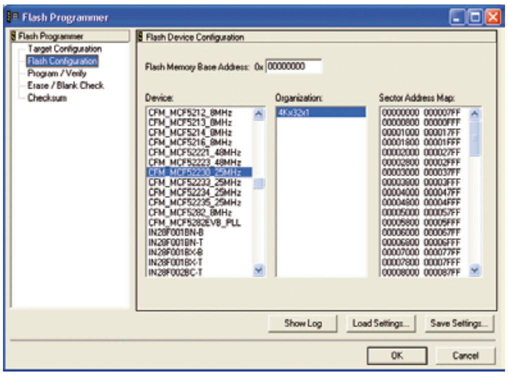
1. Fit jumper JP2 and set jumper JP1 to the right-hand position (2-3).
2. Start the CodeWarrior IDE, open the project 'elektor.mcp' using the File→Open command in the IDE menu bar. Connect the debug pod to connector J1 and apply power to the board.
3. Use Tools→Flash Programmer to bring up the Flash Programmer dialogue box.
4. In the Target Configuration menu, Browse to the Target Initialization files and there select and Open: M52235EVB_Pne.cfg
5. In the Flash Configuration menu, select: CFM_MCF52230 25 MHz.
6. Go to the Erase /Blank Check menu and run an Erase action. This is useful to check if the connection to the target is okay. If erasure is successful, the comms are okay. Click on Save Settings.
7. Now go to Program/Verify and browse to the file 'web_coordinator' (without an extension). This is a precompiled binary image, included for those who do not wish to recompile the source code.
8. Program the device! With the device programmed, the board is ready to be used.



Look in: Initialization_Files

M5208EVB.cfg
m5213evb_pne.cfg
m5271evb_pne.cfg
m5275evb_pne.cfg
M5282EVB.cfg
M5282EVB_IntROM.cfg
M52223EVB.cfg
M52235EVB_PnE.cfg
MCF523xEVB_PnE.cfg
MCF547x_Verdi.cfg
MCF547x_Verdi_Console.cfg
MCF547x_Verdi_SDRAM.cfg
MCF5206E
MCF5206eL
MCF5211.c
MCF5249C
MCF5272C
MCF5282C

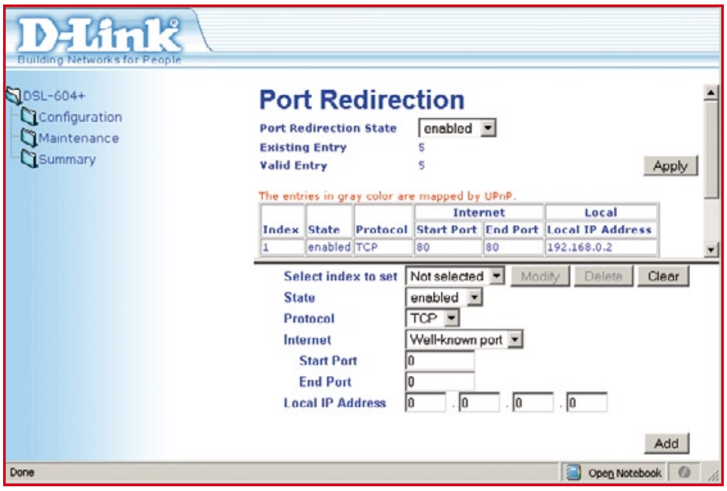
File name: M52235EVB_PnE.cfg
Files of type: Configuration Files (*.cfg)



Flash Device Configuration
Flash Memory Base Address: 0x 00000000

Device:	Organization:	Sector Address Map:
CFM_MCF5212_8MHz	4Kx32x1	00000000 000007FF
CFM_MCF5213_8MHz		00000800 00000FFF
CFM_MCF5214_8MHz		00001000 000017FF
CFM_MCF5216_8MHz		00001800 00001FFF
CFM_MCF52221_48MHz		00002000 000027FF
CFM_MCF52223_48MHz		00002800 00002FFF
CFM_MCF52230_25MHz		00003000 000037FF
CFM_MCF52233_25MHz		00003800 00003FFF
CFM_MCF52234_25MHz		00004000 000047FF
CFM_MCF52235_25MHz		00004800 00004FFF
CFM_MCF5282_8MHz		00005000 000057FF
CFM_MCF5282EVB_PLL		00005800 00005FFF
IN28F001BN-B		00006000 000067FF
IN28F001BN-T		00006800 00006FFF
IN28F001BX-B		00007000 000077FF
IN28F001BX-T		00007800 00007FFF
IN28F002BC-T		00008000 000087FF

connected to either a router or a PC. Remember that when connecting directly to a PC, the network cable must be a crossover type. Enter DigiButler's IP address in the web browser of a PC on the same network and an authorisation dialog box will be presented (**Figure 6**). The default user name is 'user' and the password is '1234567'. If required, both can be changed in the project source code.



Port Redirection

Port Redirection State: enabled
Existing Entry: 5
Valid Entry: 5

The entries in gray color are mapped by UPnP.

Index	State	Protocol	Internet Start Port	Internet End Port	Local IP Address
1	enabled	TCP	80	80	192.168.0.2

Select index to set: Not selected
State: enabled
Protocol: TCP
Internet: Well-known port
Start Port: 0
End Port: 0
Local IP Address: 0 . 0 . 0 . 0

Figure 5. Most routers support port redirection or port forwarding. Setting the IP address and port number allows Internet users to access DigiButler on a local network.

Successful authentication brings up the default main web page, *index.htm*. As illustrated in **Figure 7**, a graphic image on the example default web page shows the status of the relay. The project source code allows you to substitute your own image! A dark image signifies that the relay (RE1) (lamp) is off, while a bright image shows that the relay (lamp) is on. The relay can be switched on and off remotely using Light ON / Light OFF button on the web page. Application note AN3455 (ref. [2] in part 1) gives an in depth description of an alternative method of switching the relay on and off using web page variables in a browser address bar. From the same web page it is also possible to set a timer for when the relay should switch on or off, this function making use of the Coldfire's real time clock. And lastly, the example page allows the user password to be changed if desired.

The process for talking to DigiButler over the Internet is exactly the same. Entering the router IP address in the browser of an Internet connected PC gives the same results, but only if the router has been set up for forwarding port 80 to the DigiButler's local IP address. In our example, the router address is 86.131.22.120. Be aware that it might not be possible to access the server from a PC within a local network, if you use the router IP address.If you are having difficulty remembering the IP address, NO-IP [3] offers a free service which maps the IP address to an easy to remember sub domain name.

Finding an Internet connected PC is not always possible or safe, so WAP phone access is also supported. Nearly all modern mobile phones have WAP, but the service is subject to what your provider is offering you. DigiButler's default WAP page is *wap.wml*. Entering our example URL,

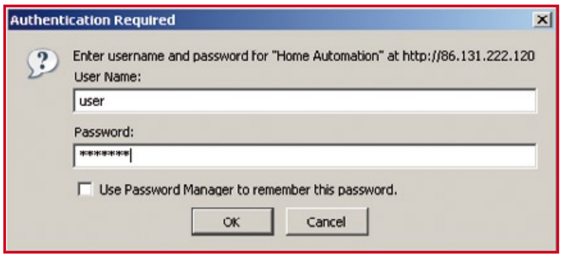
http://86.131.22.120/wap.wml

from the browser of a WAP enabled phone, gives similar results as before but without the images. The WAP pages have been tested using Nokia and Samsung mobiles, but may need to be changed to support other manufacturers' handsets.

Creating and uploading web pages

A neat feature of the software is its support for dynamic uploading of new web pages. The following is intended for those having some experience in programming web-connected devices.

By implementing an Ethernet writeable FLASH based file system, it's possible to update web pages remotely without reprogramming the Coldfire device. Two files systems are actually implemented — a static compile time system and the dynamic writable system. Both can be used to store web pages, graphics files and other content and both allow directory structures and file names up to 255 characters. Since the application looks in the writable system for a requested file before searching the compile time file system, it's possible to override the compiled files. For example, if an image file exists in both file systems with the same name, then the file in the writeable system will be used and the one resulting from the compile will be ignored.



Enter username and password for "Home Automation" at http://86.131.222.120
User Name: user
Password: *******
☐ Use Password Manager to remember this password.

Figure 6. The first access in a web session requires the user to enter a username and password before access is granted to the main page of the home automation server.

Example web pages and Ethernet loading utilities can be found in the *web_page* directory of the project. The files required to upload new web content to the automation server are *make.bat*, *filelist.txt*, *emg_dynamic_ffs.exe* and *emg_web_uploader.exe*.

The batch file *make.bat* conveniently calls the necessary utilities to compress and upload the web-resultant content image file using the files listed in *filelist.txt*. Before executing *make.bat*, change the IP address in the line

```
emg_web_uploader 10.171.88.63 dy-
namic.ffs joshua
```

in *make.bat* to the board's IP address as shown in the terminal window. The parameter *dynamic.ffs* is the name of the compressed file image generated by compression utility *emg_dynamic_ffs.exe*. This compresses the files in *filelist.txt* into a single image. The parameter *joshua* is the upload security key, used to prevent unauthorized upload of web content. DigiButler will verify the supplied security key with the version in memory and upload only if both match. The security key can be changed in the server source code if required.

An example of the contents of *filelist.txt* is shown below. These files will be compressed into one image for upload to the server. The last line in the file must be a blank line with only a CRLF (i.e. hit Enter in the last blank line).

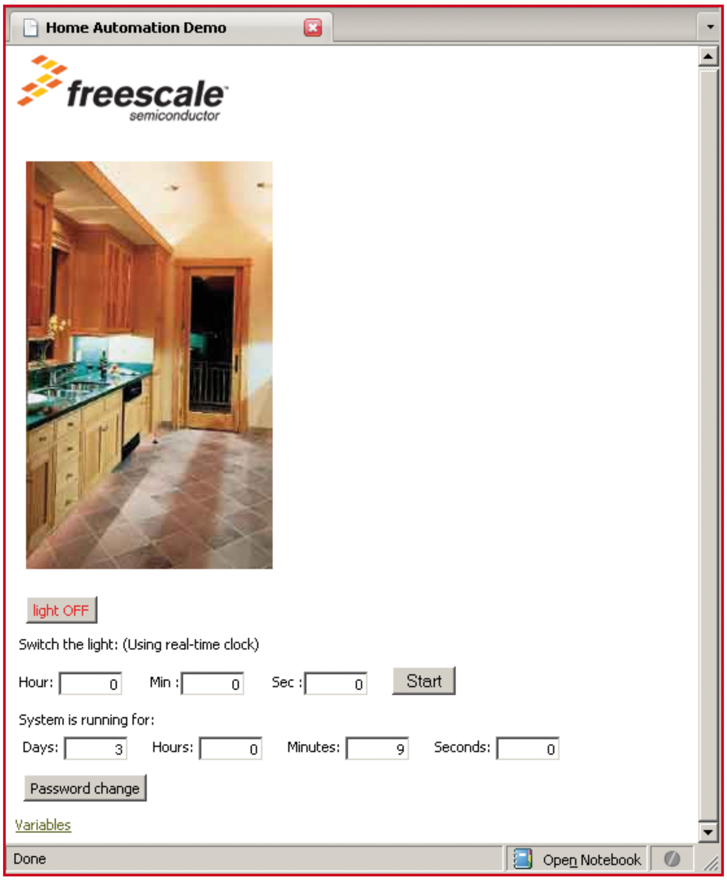
```
* This is a list of the files
stored in the bin file.
```

```
* The files are arranged
in the order they are
listed here.

index.htm
kitchen.jpg
black.jpg
main.js
pot_data.txt
fsllogo.gif
variables.htm
wap.wml
wap1.wml
wap2.wml
wap_main.wml
incorrect.wml
```

Overcoming firewalls

Both FTP or TFTP protocols are commonly used to upload files to webservers. However, many firewalls and routers consider these protocols as dangerous and block the transfer, which causes a problem if trying to upload new content to the automation server. To overcome this, the utility *emg_web_uploader.exe* uses TCP and port 80 thereby getting around the problem.



Switch the light: (Using real-time clock)

Hour: 0 Min : 0 Sec : 0

System is running for:

Days: 3 Hours: 0 Minutes: 9 Seconds: 0

Figure 7. The first web page the user sees when accessing DigiButler from a PC web browser. You can switch the kitchen lighting on and off.

Enhancements and your contributions

DigiButler opens up many exciting opportunities for controlling devices remotely and for remote real time data measurement. Analogue and digital signals can be captured, measured and subsequently analysed from a web browser. For example, a sensor could be added to the DigiButler analogue inputs and be read remotely using the HTML VARS function described in AN3455. With some modification to the software, images could be captured using a VGA camera module such as the C328-7640 [4].

The prototyping area on the board allows you build your own extension circuits. Many signals to and from the microcontroller are conveniently found on J11 and J12, while the V_{ext} pin close to the fuse may help to power the extension(s). The same for the +3.3 V rail which can be 'tapped' near J11 (but watch the extra current consumption of your extension circuit). If you duplicate the existing relay circuit for the control of a second 230 VAC load, be sure to observe electrical safety precautions.

The authors look forward to seeing your applications and extensions as the project was designed and published with the open-source concept in mind. A dedicated web page will be 'at your service' at www.elektor.com/digibutler_en.

(071102-1)

Web Links

[1] www.freescale.com/webapp/sps/site/prod_summary.jsp?code=CWS-MCF-STDED-CX&nodeId=0127269401 2E704080&fpsp=1&tab=Design_Tools_Tab

[2] www.pemicro.com/products/product_viewDetails.cfm?product_id=163&menu_id=details&CFID=1032336&CFTOKEN=52342289

[3] www.no-ip.com/services/managed_dns/free_dynamic_dns.html

[4] VGA camera module
www.electronics123.com/s.nl/it.A/id.2027/.f?sc=8&category=233

Changing the IP address

You may wish to use a fixed IP address instead of having an IP address assigned by the router's DHCP server functionality. This is possible by making two simple code changes. The first is to switch off the DHCP functionality by commenting out the line

```
#define DHCP_CLIENT      1  /* include DHCP client code */
```

in the file 'ipport.h'. The second is to set the IP and gateway addresses and subnet mask. Example values for these parameters are included in file 'main.c'. Change them according to your setup and remember to change the #if 0 statement to #if 1.

```
#if 0
IP_ADDRESS(213,194,223,49);
DEF_GATEWAY(213,194,223,1);
SUBN_MASK(255,255,255,128);
```