



APPLICATION NOTE 2155

DS80C400 Software PHY Reset

Abstract: There is a potential timing issue causing problems with the Ethernet receiver when a physical interface (PHY) such as the Intel® LXT972ALC is connected to the reset output (RSTOL) of the DS80C400. The solution is to not connect the reset line and perform a software reset of the PHY instead. This application note describes the code necessary to perform a software reset of the PHY.

Introduction

The [DS80C400](#) high-speed microcontroller has a built-in Ethernet media-access controller (MAC) with an industry-standard media-independent interface (MII). Please refer to the DS80C400 data sheet and the [High-Speed Microcontroller User's Guide: Network Microcontroller Supplement](#).

There is a potential timing issue causing problems with the Ethernet receiver when a physical interface (PHY) such as the Intel LXT972ALC is connected to the reset output (RSTOL) of the DS80C400. The solution is to not connect the reset line but instead perform a software reset of the PHY. This application note describes the code necessary to perform a software reset of the PHY.

DS80C400 MAC Software Library

Code to access the DS80C400 MAC and external PHYs is available on our ftp site at <http://files.dalsemi.com/tini/ds80c400/ethdriver/> and documented in application note 712, "[DS80C400 Ethernet Drivers](#)."

PHY Software Reset Procedure

The IEEE® Standard 802.3 defines bit 15 in the MII control register (0.15) as the reset bit. The bit is self-clearing. Using the aforementioned library, a PHY can be reset with the following code:

```
mov     b, #0                ; PHY number
mov     a, #MII_CONTROL      ; Control register
call    ETH_ReadMII
orl     l, #80h              ; Set reset bit (0.15) in r1
call    ETH_WriteMII

reset_wait:
call    ETH_ReadMII
mov     a, r1
jnb     acc.7, reset_wait    ; Loop while in reset
```

Equivalent code can be written for Java™ programs running on the TINI® Firmware 1.12 and later. `com.dalsemi.tininet.TININet` contains the `readMII` and `writeMII` methods:

```
import com.dalsemi.tininet.*;

...
```

```
int val = TININet.readMII(0, 0);          /* PHY 0, register 0 */
val |= 0x8000;                          /* Set reset bit */
TININet.writeMII(0, 0, val);            /* Perform reset */
do {
    val = TININet.readMII(0, 0);
} while ((val & 0x8000) != 0);
```

IEEE is a registered service mark of the Institute of Electrical and Electronics Engineers.

Intel is a registered trademark of Intel Corporation.

Java is a trademark of Sun Microsystems, Inc.

TINI is a registered trademark of Maxim Integrated Products, Inc.

Application Note 2155: www.maxim-ic.com/an2155

More Information

For technical support: www.maxim-ic.com/support

For samples: www.maxim-ic.com/samples

Other questions and comments: www.maxim-ic.com/contact

Automatic Updates

Would you like to be automatically notified when new application notes are published in your areas of interest?

[Sign up for EE-Mail™.](#)

Related Parts

DS80C400: [QuickView](#) -- [Full \(PDF\) Data Sheet](#) -- [Free Samples](#)

DSTINIs400: [QuickView](#) -- [Full \(PDF\) Data Sheet](#)

AN2155, AN 2155, APP2155, Appnote2155, Appnote 2155

Copyright © by Maxim Integrated Products

Additional legal notices: www.maxim-ic.com/legal