Z i L O G

UP007801-1205

Product Update

Errata for Z02201

TRADEMARKS

Z8®, Z02201™, and Z02205™ are trademarks of ZiLOG, Inc.

DATA PUMP CHIP RELEASES

Data Pump Version	ROM code	Earliest date code: yyww (ww=week)
Z02201 Ver. 0x31	R3470	Use 9801 and later
Z02201 Ver. 0x48	R4078	9848 (released Jan'99)

The following fdata pump firmware enhancements made to Version 0x31 are now available in Version 0x48 of the Z02201:

- Added simultaneous tone detection and generation (Config.MODE = 4)
- Updated the V.23 code base
- Corrected the notch filter equation for **Alpha**
- Fixed the timing issue with pulse dialing
- Increased the FSK for better noise-immunity
- Improved Connectivity

> **Note:** Problems numbered 1–7 and 13 are corrected and must be removed from the user's controller code when using Ver 0x48 of the code. The only known problem encountered with Version 0x48 is included as Problem #14 in this document.

The following paragraphs describe some workarounds and typical issues when using the Z02201.

Problem #1: DTMF Tone Generation

Versions:	No release versions of the Z02201
Updated:	24-June-98

Problem: The 04-Mar-98 errata sheet erroneously reported that DTMF tones were transmitted about 4 dBm lower than expected in the Z02201 version `0x31` (R3470) data pumps. This problem occurs only in the Z02201 `0x27` and earlier versions, because the data pumps are not distributed.

Problem #2: Discrete Tone Detection Below -35 dBm

Versions:	Z02201 Ver. 0x31 (R3470).
Updated:	24-June-98

Problem: The discrete tone detectors do not continuously detect their programmed frequencies at low received signal levels (below -35 dBm). The tone detector status bits can indicate no tone is detected when the tone is present. The status bits also can indicate a tone is detected when the tone is not present.

Solution: Program biquad tone detector B to detect the presence of at least -43 dBm power in the answer tone frequency range (1650–2250 Hz). whenever a discrete tone detector status bit detects a desired tone, check the biquad B power level to determine when to ignore the status bit turning off. To protect against false detection, enforce an experimentally determined minimum `On` time for the tone detector status bit. The `On` time varies with the frequency detected. Experimentation found a -43 dBm V.25 answer tone (2100 Hz) to be detectable for 150 ms continuously within a 3.3 second detection period, and a -43 dBm Bell answer tone (2225 Hz) to be detectable for 30 ms continuously within a 3.3 second detection period.

Suggested coefficients for biquad tone detector B are represented in the following table.

B2	B1	A3	A2	A1
0xd136	0x3396	0x0000	0x0000	0x19f2
0xcf8e	0x08bc	0x0000	0x0000	0x19f2

The Biquad tone detector B power level is read from DSP RAM location 0x4e (the 04-Mar-98 errata sheet reported this location incorrectly). When using the telephone line interface (that is, data access arrangement, or DAA) from the revision 3 Z02205 demo boards, the following condition holds true: if the value read is greater than or equal to `0x60,` then the biquad tone detector B detects power greater than or equal to -43 dBm in its programmed frequency range. If a different telephone line interface is used, determine if the biquad tone detector B value corresponds to -43 dBm.

Addendum 24-Jun-98: To detect Bell answer tone (2225 Hz) at power levels below -35 dBm, an alternate method to using a 30 ms detection period is to require continuous detection for 100 ms of any of these 4 answer tones: 1650 Hz (V.21 marks), 2100 Hz (V.25 answer tone), 2225 Hz (Bell answer tone) or 2250 Hz (V.22 marks). At low power levels the discrete tone detectors detect a Bell answer tone as one of the 4 frequencies for sufficient time to allow this method to operate.

Problem #3: Guard Tone Generation

Versions:	Z02201 Ver. 0x31 (R3470).
Updated:	04-Mar-98

Problem: Guard tones are transmitted with incorrect power.

Solution: Set the guard tone transmit power before it is transmitted. RAM location `0x186` is set to the guard tone transmit level. The guard tone power level is normally set to 3 dBm below the data mode transmit level (`TxLevel`). The formula to determine the correct value is the same as for `TxLevel (10^(lev/20)*2048).`

Problem #4: V.22bis/V.22/Bell212A Training

Versions:	Z02201 Ver 0x31 (R3470).
Updated:	04-Mar-98

Problem: In V.22bis/V.22/Bell212A modes, the receiver sometimes does not finish training by the time the transmitter has finished (that is, when `R5.CDET = 1`). In this case it is not safe to enter data mode.

Solution: Retrain, if possible, in V.22bis; otherwise, abort the connection. Perform the following steps:

1. Read RAM location `0x0F9` and check the least significant 8 bits. If the bits are non-zero, the on-line transferring of data is now possible. If the least significant 8 bits are 0, the receiver did not finish training. Proceed to step 2.

2. Check the CONN_Mode RAM location (0x1F0). If the value of the least significant 8 bits is 8, the user can retrain. If it is not 8, the connection attempt has failed. Hang up and try again.
3. To retrain, the user must first enable retraining (in essence, the data pump still thinks it is handshaking and retrains are not normally allowed during the handshake phase). To enable retraining, set RAM location 0x0F4 to the value 0x38C.
4. From this point, the user can now perform the retrain normally (set the Mstatus.RETRAIN = 1).
5. When R5.CDET = 1, the reatrain has finished, and the transfer of data is now possible.

Problem #5: V.21/V.23 Training

Versions:	Z02201 Ver. 0x31 (R3470).
Updated:	04-Mar-98

Problem: In V.21/V.23 modes, when a V.25 answer tone is used, the calling modem can detect the 2100 Hz V.25 answer tone as carrier for a brief period.

Solution: Debounce an R5.CDET = 1 indication for 0.2 seconds before declaring the handshake complete.

Problem #6: Timed DTMF Dialing

Versions:	Z02201 Ver. 0x31 (R3470). Corrected in Z02201 Ver. 0X48 (R4078).
Updated:	24-Jun-98

Problem: In timed DTMF dial mode, the DPBUSY bit does not get set by the data pump at the beginning of the tone generation, yet resets properly at the end.

Solution: Force the DPBUSY bit to 1 after the data pump is told to start the DTMF tone (RTSP = 1). This action is accomplished by setting bit 3 of DSP RAM location Mstatus (0x1F7). For Example.

```
/* * Tone dial a digit. Addendum 24-Jun-98: Assumes that Config.MODE = 3
 * (dial mode) and the timing parameters have also been set.
 */
void tone_dial(digit)
    char digit;
{
    ltemp = read_DSP_RAM(DTMFREG);    /* set the digit */
    ltemp &= 0xfff0;
    ltemp |= digit | TONEDIAL;        /* set tone mode (b7=1) */
    write_DSP_RAM(DTMFREG, ltemp);
    RTS_on();                         /* transmit the tone (set RTSP = 1)*/
    /*
     * set b3 of 0x1f7 -- bug workaround for V31 (R3470) of DSP code
     * -- this sets DPBUSY = 1
     */
    write_DSP_RAM(0x1f7, read_DSP_RAM(0x1f7) | 0x8);
    while (!(inp(MDMST) & DPBUSY));   /* wait for start */
    while (inp(MDMST) & DPBUSY);      /* wait for completion */
    RTS_off();                        /* turn off tone */
}
```

Problem #7: Carrier Detection in V.22bis, V.22, and Bell 212A is at a -42 dBm Receive Level and Lower

Version:	Z02201 Ver. 0x31 (R3470).
Updated:	23-Dec-98

Problem: At receive levels of -42 dBm and below, the data pump may set `R5.CDET = 0` for brief periods in V.22bis, V.22, and Bell 212A data modes. These brief carrier drops may cause a modem to hang up the telephone line even though the connection could otherwise pass data. Also, because the data pump clamps received data to the marking state (all 1 bits) while `R5.CDET = 0`, a greater number of data errors occur at these low receive levels even if the modem does not hang up the telephone line.

Note: *23-Dec-98:* Errata sheets prior to this date contained an error in the routine DisableCDET() that could cause data errors during V.22 or Bell 212a operation (that is, 1200 bps) on clear connections. The one changed line in the code sample starting on page 5 is marked with a comment.

Solution: Disable the data pump's carrier power calculation used in V.22bis, V.22, and Bell 212A. In these data modes, read the carrier power from data pump RAM, and compare it to the desired `RLSD On` threshold (for example, a constant or the value from the data pump's `RLSDOnThresh` RAM location). If the carrier power drops below this threshold, re-enable the data pump's carrier power calculation and use `R5.CDET` to determine if the telephone line should be hung up. Perform these steps to use the sample `C` code starting on page 5.

1. In V.22bis, V.22, and Bell 212A data modes, call `DisableCDET()` after `R5.CDET` indicates the connection has been established, the modem is transferring data, each retrain completes, and the carrier has been restored following a carrier loss of short duration (that is, not resulting in a hang up).

2. In V.22bis, V.22, and Bell 212A data modes, call `EnableCDET()` immediately before setting `Config.MODE = 0` (standby) after data mode operation, and during data mode operation when either `R5.CDET` indicates any carrier loss (including retrain) or CarrierPower() indicates there is insufficient carrier power.

3. Call `CarrierPower()` during data mode operation when the modem is transferring data if `DisableCDET()` was called to disable the data pump's `R5.CDET` bit. If `CarrierPower()` returns FALSE, call `EnableCDET()` to enable the data pump's carrier power calculation, then use `R5.CDET` to debounce the carrier loss to determine if the telephone line should be hung up.

Note: The data pump's R5.CDET bit is used for several purposes: insufficient carrier power, handshaking in progress, retrain in progress, and hang-up. R5.CDET is used as an indicator of handshaking in progress, retrain in progress, and hang-up during this workaround (only the carrier power detection is disabled).

```
/*
 * Disable the data pump's carrier power calculation used in V.22bis, V.22 and
 * Bell 212A. The result of this calculation is normally reflected in the data
 * pump's R5.CDET bit.
 *   Call this routine after R5.CDET indicates the modem is transferring data
 * (i.e. after R5.CDET indicates the connection has been established), after
 * each retrain has completed, and after carrier has been restored following
 * a carrier loss of short duration (not resulting in hang up).
 * Call EnableCDET() and CarrierPower() as indicated to ensure the R5.CDET
 * bit is active when needed.
 */
void DisableCDET(void) {
UINT16 ltemp; /* Unsigned 16-bit integer */
/* Disable the carrier power calculation */
write_DSP_RAM(0xe8, 0x327);
/* Set two data pump RAM locations to force the output of the carrier power
 * calculation to indicate there is carrier power.
 */
ltemp = read_DSP_RAM(0x1fb);
ltemp |= 0x4000;
write_DSP_RAM(0x1fb, ltemp);
ltemp = read_DSP_RAM(0x1fa); /* This line revised 23-Dec-98 */
ltemp |= 0x4000;
write_DSP_RAM(0x1fa, ltemp);
}
/*
 * Enable the data pump's R5.CDET bit to reflect the status of the carrier.
 * Call this immediately before setting Config.MODE = 0 (standby) after data
 * mode operation, and after CarrierPower() indicates there is insufficient
 * carrier power.
 */
void EnableCDET(void) {
write_DSP_RAM(0xe8, 0xbc4);}
/*
 * Read the carrier power from the data pump RAM, if it is above the RLSD ON
 * threshold return TRUE, otherwise return FALSE. Call this routine during
 * connections when the modem is transferring data if DisableCDET() was called
 * to disable the data pump's R5.CDET bit. If this routine returns FALSE, call
 * EnableCDET() to enable the data pump's carrier power calculation before
 * using R5.CDET to determine if the telephone line should be hung up.
 */
UINT8 CarrierPower (void) {
UINT8 scaler;      /* Unsigned 8-bit integer */
UINT16 ltemp; /* Unsigned 16-bit integer */
ltemp = read_DSP_RAM(0x195); /* Read the carrier power */
scaler = ((read_DSP_RAM(0x1f7) >> 8) & 0x30); /* Get the scale bits */
if (scaler == 0x30) {/* If both bits are on */
ltemp >>= 4;   /* divide the power reading by 16. */
} else if (scaler) {/* If just one of the bits is on */
ltemp >>= 2; /* divide the power reading by 4. */
}
if (ltemp >= RLSDON) {/* Compare carrier power to RLSD On threshold */
return TRUE; /* Return, indicating sufficient power. */
}
return FALSE; /* Return, indicating insufficient power. */
}
```

Problem #8: Cosine in DTD0-DTD15 Formula

Versions:	All versions of the Z02201 added into the Product Specification.
Updated:	24-Jun-98

Note: The description of data pump RAM locations DTD0-DTD15, contains this formula using trigonometric cosine:

where f_{tone} is the frequency of the tone to be detected.

$$coeff_{tone} = \cos(\frac{2\pi \cdot f_{tone}}{9600}) \cdot 32767$$

Problem: The value within the brackets in the formula represented above cannot be calculated in degrees.

Solution: Calculate this cos() in radians–not degrees.

Problem #9: DTMF Power Levels

Versions:	All versions of the Z02201 added into the Product Specification
Updated:	24-Jun-98

Problem: The Z02201 and Z02922 data pumps have limited maximum output power. This condition applies not only to data mode, but also to DTMF and other tone generation. During DTMF or tone generation, if the sum of the transmit levels programmed into `DTMFh_lev` and `DTMFl_lev` exceeds `30720 (0x7800)` the data pump may not properly transmit the tones.

Solution: When transmitting DTMF tones with a required twist (power difference between high and low bands), use this formula to determine the maximum DTMF transmit levels:

`10^(x/20) + 10^((x+b)/20) <= 30720/32768`

where x is the DTMF low band (`DTMFl_lev`) transmit level in dBm,

`x+b`is the DTMF high band (`DTMFh_lev`) transmit level in dBm (`b` is the twist in dBm).

The following table represents values for maximum transmit levels (DTMFl_lev + DTMFh_lev = 30720) at common twist values:

DTMFl_lev	DTMFh_lev	x	x+b	b
14,477	16,243	-7.10	-6.10	1
13,599	17,121	-7.64	-5.64	2
12,733	17,987	-8.21	-5.21	3

Problem #10: V.29 Receive

Not applicable.

Problem #11: HDLC Status During V.29 Receive

Not applicable.

Problem #12: Data Pump Waking Up From Sleep Mode

Versions:	All versions of the Z02201
Updated:	24-Jun-98

Problem: The Z02201 data pump requires between 5 and 6 milliseconds to wake up from `SLEEP` mode. This condition is longer than the time shown in other documentation. While waking up, if the host controller writes to the data pump, the writeis ignored; however, the host still interprets that the action is completed.

Solution: After writing the data pump register `DataP` to wake the data pump, the host controller should either wait 10 milliseconds for the data pump to wake up, or should examine the `R5.RES` bit until it has changed to 1. From that point, the bit can be changed back to 0.

Problem #13: Pulse Dialing Timing Inaccuracy

Versions:	All versions of the the Z02201 prior to Ver. 0X48 (R4078). Corrected in Z02201 Ver. 0X48 (R4078).
Updated:	23-Dec-98

Problem: The data pump performs pulse dialing by toggling the $\overline{\text{OH}}$ pin. The data pump also clocks changes to the $\overline{\text{OH}}$ pin using the serial receive clock. Since the default value of the serial receive clock upon reset is 110 Hz, the rate at which the $\overline{\text{OH}}$ pin can be changed after reset is limited to approximately once every 9 ms.

Solution: Increase the serial receive clock rate by putting the data pump into V.22bis data mode for a brief period before beginning to dial. Note that the transmitter must be squelched before putting the data pump into V.22bis data mode to prevent the data pump from transmitting. The solution may be implemented following these steps immediately before having the data pump generate a pulse dial:

1. Set `Config.MODE = 0` (standby)
2. Set `DpCtrl.TXSQLCH = 1` to squelch the transmitter.
3. Set `Config = 0x4408`, then set `TrnCtrl = 7` to instruct the data pump to enter V.22bis data mode using a manual handshake. The data pump transmits nothing since the transmitter is squelched.
4. Wait 5 ms to ensure the data pump enters V.22bis data mode and begins transmitting a squelched (or silent) tone. When this transmission occurs the serial receive clock is changed to 2400 Hz, allowing the $\overline{\text{OH}}$ pin to be toggled approximately every 0.4 ms.
5. Set `Config.MODE = 0` (standby), and follow the procedures required to make the data pump pulse dial.

Problem #14: EQMLev and DpCtrl.EQE

Versions:	Z02201 version Ver. 0X48 (R4078).
Updated:	19-Feb-99

Problem: The data pump reports different values in RAM location `EQMLev` than earlier data pump versions. This report causes the modem controllers that monitor `EQMLev` for conditions of high noise to hang up, retrain to hang up, or retrain unnecessarily. Also, the `DpCtrl.EQE` RAM location does not operate because its value is based on the `EQMLev` value being greater than the `EQMMaxThresh` value. As described below, the noisy line test for this version of the data pump is to consider `EQMLev` to be a signed 16-bit integer, and test to see if it is less than a data mode-specific threshold value.

Solution: During V.22bis (2400 bps) operation, modify the high noise hang up test as follows: the line is too noisy if:

```
(INT16) EQMLev < (INT16) -2048
```

During V.22 or Bell 212 (1200 bps) operation, modify the high noise hang up test as follows: the line is too noisy if:

```
(INT16) EQMLev < (INT16) -2816
```

During V.22bis operation, modify the retrain test as follows: the line is too noisy if:

```
(INT16) EQMLev < (INT16) -1024
```

Revision History

04-Mar-98	Problems 1-6	Added.
24-Jun-98	Problem 1	Removed. The problem does not occur in any Z02201 data pumps being distributed.
24-Jun-98	Problem 2	Revised. Corrected the address of the data pump RAM location containing the Biquad tone detector B’s power level. Also added an alternate method of detecting a Bell answer tone at low receive levels.
24-Jun-98	Problem 6	Revised. Added a note in the ‘`C`’ code example, describing the data pump initialization done before the code is executed.
24-Jun-98	Problems 7-12	Added.
21-Jul-98	Problems 2-7, 10, 11	Removed references to firmware versions not commonly available in chip form. Corrected a typo identifying problems 10a & 10b as 11a & 11b. Since problem descriptions and solutions were not modified, the ‘Updated’ dates on these problems were not changed.
23-Dec-98	Problem 7	Corrected an error in the routine DisableCDET() that could cause data errors during V.22 or Bell 212a operation (that is, 1200 bps) on clear connections. The one changed line in Table 2 is marked with a comment.
23-Dec-98	Problem 13	Added.
22-Feb-99	Problem 14	Added.

This publication is subject to replacement by a later edition. To determine whether a later edition exists, or to request copies of publications, contact:

ZiLOG Worldwide Headquarters
532 Race Street
San Jose, CA 95126
Telephone: 408.558.8500
Fax: 408.558.8300
www.ZiLOG.com

Document Disclaimer

ZiLOG is a registered trademark of ZiLOG Inc. in the United States and in other countries. All other products and/or service names mentioned herein may be trademarks of the companies with which they are associated.

Information in this publication concerning the devices, applications, or technology described is intended to suggest possible uses and may be superseded. ZiLOG, INC. DOES NOT ASSUME LIABILITY FOR OR PROVIDE A REPRESENTATION OF ACCURACY OF THE INFORMATION, DEVICES, OR TECHNOLOGY DESCRIBED IN THIS DOCUMENT. ZiLOG ALSO DOES NOT ASSUME LIABILITY FOR INTELLECTUAL PROPERTY INFRINGEMENT RELATED IN ANY MANNER TO USE OF INFORMATION, DEVICES, OR TECHNOLOGY DESCRIBED HEREIN OR OTHERWISE. Except with the express written approval of ZiLOG, use of information, devices, or technology as critical components of life support systems is not authorized. No licenses are conveyed, implicitly or otherwise, by this document under any intellectual property rights.