Application Note

CMX865A in Short Local Loops

Publication: AN/Telecom/SLL/865A /4 April 2009

1 Introduction

Fixed wireless terminals, also known as wireless local loops, provide a simple means of interfacing existing POTS devices, like telephones or fax machines, to cellular networks. To achieve this simple connection, the terminal must regenerate a POTS line locally: the short local loop. Because this loop is very short and does not extend beyond the premises, the driving and signalling requirements are relatively simple. This application note suggests how such a loop may be implemented using the CMX865A Telecom Signalling Device and the Silver Telecom AG1171.

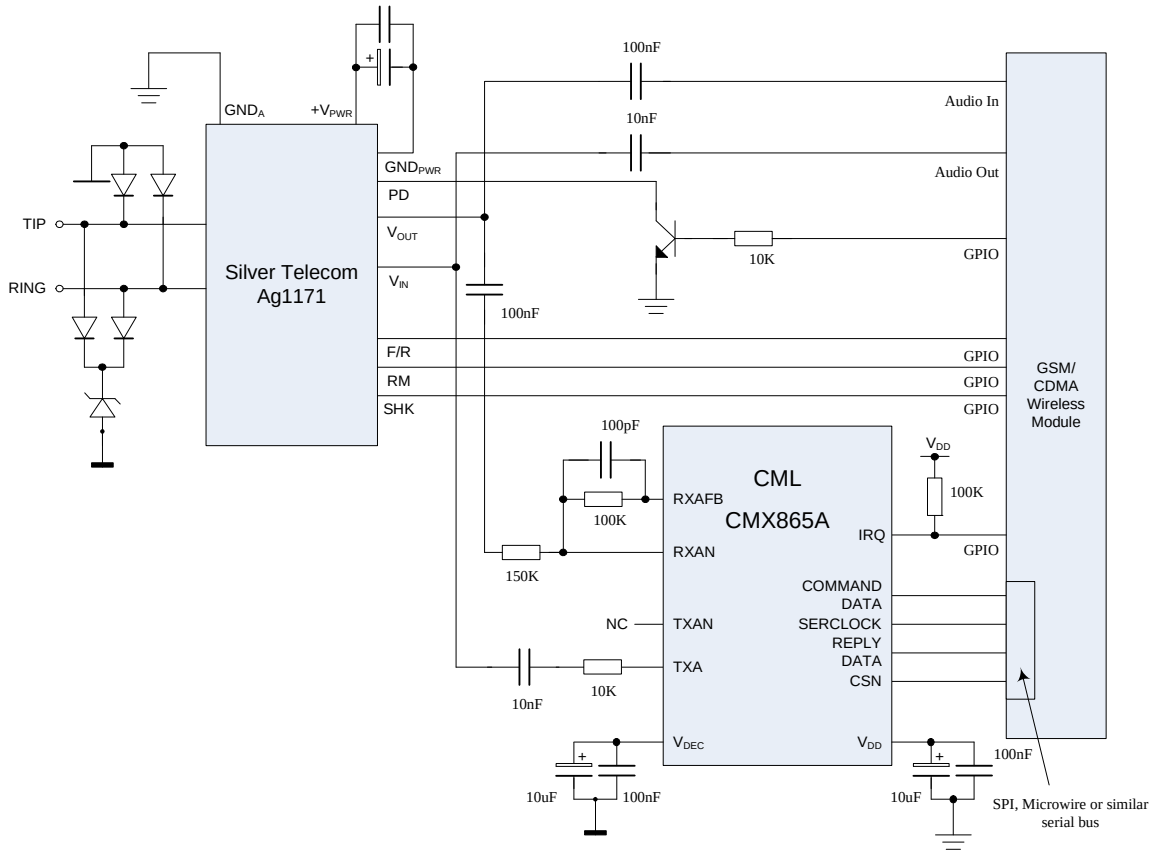


Figure 1. Simplified fixed wireless terminal, wireless local loop application showing typical connections

2 Line Regeneration

The Silver Telecom AG1171 provides a low-cost method of regenerating a short local loop but does not provide all of the signalling required. Much of this will come from the far exchange via the audio path of the cellular link but, as the audio path is only established once a call has been accepted, some signalling must be regenerated. The AG1171 also provides the -48V DC supply without which phone equipment will not operate.

The AG1170 is able to determine the hook status of connected equipment and to provide line reversals or a pseudo-ringing signal using a series of controlled line reversals. Re-generation of call progress tones, Calling Party Identification (Bellcore version of type 1 caller-ID signalling) and DTMF decoding are provided by the CMX865A.

3 Call Progress Tones

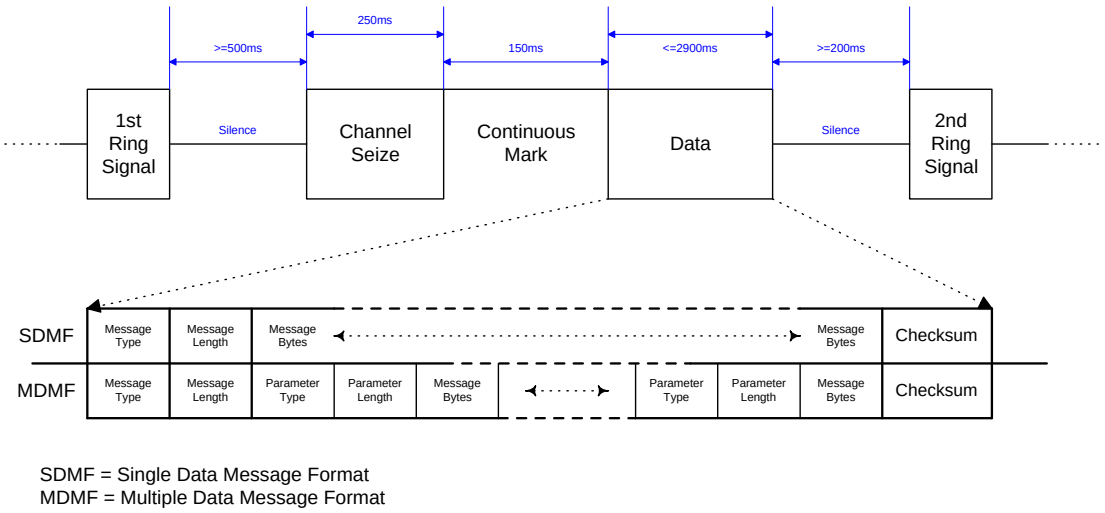
There are a large number of call progress tones in common usage so the CMX865A provides a tone generator block that can be programmed with a subset of those required. There are four registers for the transmission of tones and, if used, these must be programmed whenever the CMX865A is power cycled or reset. Programming information is given in the CMX865A Data Sheet and programming methods are given in other CML Application Notes. For illustration, the following sections assume;

1. Dial tone is 350 + 440 Hz continuous.
2. “Error tone” is 350 + 440 Hz cadenced.
3. Ring tone is 400 + 450 Hz cadenced.
4. Ringing signal is 17.1Hz.

The first three tones listed will be programmed into the Tone pair registers TA, TB and TC respectively.
The dial tone is required to signal to that the POTS line is not in use AND that the cellular device is able to make an outgoing call.
An “error tone” is used generically here to cover standard network call progress tones such as ‘busy’ (engaged), ‘congestion’, ‘information’ and also when the cellular device is not responding or the cellular network is unavailable. It is expected that the designer will program the relevant tones for each condition.
Ring tone is the tone that is produced in the calling parties handset to indicate that the called parties line is ringing.
Ringing signal is the high amplitude signal that causes POTS connected equipment to audibly ring (if a ringing device is fitted). This signal is not available on the CMX865A and is generated by the AG1171.

4 Caller ID

The caller-ID packet format illustrated is Bell SDMF format and is referred to as CLI, in line with Bell terminology. Figure 2 illustrates the line signalling and data format for this Bellcore CLI scheme, upon which many caller-ID applications are based.



SDMF = Single Data Message Format
MDMF = Multiple Data Message Format

Figure 2. Bellcore On-hook CLI System Signals and Data Format

Notes:

1. In the Bellcore scheme, the CLI is preceded by a ringing signal that is generated by the AG1171, the cadence being controlled by the host processor.

2. If no CLI is present on a Bellcore system then the silence period between ringing signals may extend for 4 seconds (nominal).

5 General Information

1 Interface details, line level specifications and other technical requirements are not covered because these are dependent on the application and the intended country of operation. For this information please contact your local Service Provider or refer to the relevant specifications.

2 The following syntax is used to describe the CMX865A C-BUS registers.
- a *General Control* ($E0) - The register name is italicized followed by the hexadecimal address of the register in brackets.
- b b2 = 1 - The register bit, bit 2, is set to a 1.
- c b5..b2 = 1001 - The register bits, bit 5 through bit 2, are set to 1001 respectively.

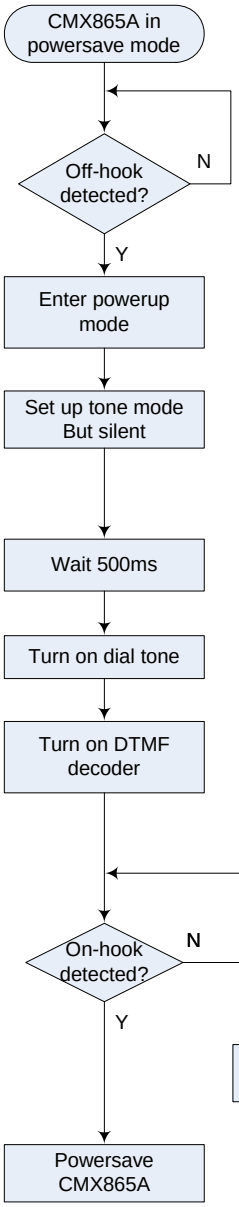
3 Figures 4 and 7 both assume that the CMX865A is powered up and a General Reset Command, C-BUS address $01 (no data), has been issued. The CMX865A will, therefore, be in powersave mode with registers in their default states.

4 CMX865A registers are either write or read -only and not bit addressable. It is suggested that a shadow (working copy) of each register is maintained in the host so that bit settings can be changed by a read-shadow, mask and write operation. A shadow of the Status register should be maintained in the host so that status bits are not unexpectedly lost because they are cleared following the read.

5 Additional timers and error handling may be required.

6 It is assumed that interrupts or polling will start some processes. For clarity, links between these have been left out.

7 Bit changes are given when the remainder of the register may need to be preserved.



The CMX865A is powered up but in powersave mode.

An off-hook event is signalled to the host by the AG1171: SHK = 1

Bring the CMX865A out of powersave mode.
Write *General Control* ($E0) = $8100 //TXAN tri-state, powerup high
//TXA will automatically go to BIAS

Set-up user programmed tones.

Write *Transmit Mode* ($E1):
b15..12 = 0001 //select Tx DTMF/tone mode
b11..9 = *application dependent* //select Tx level
b4 = 0 //select tone mode
b3..0 = 0000 //no tone

Arbitrary settling delay before starting dial tone. The length of this delay may be subject to local regulations.
This settling delay also satisfies the recommended 20ms delay for the CMX865A and crystal to stabilise.

Write *Transmit Mode* ($E1):
b3..0 = 1100 //enable user-define tone TA

Write *Receive Mode* ($E2):
b15..12 = 0001 //select Rx tone detect mode
b11..9 = *application dependent* //select Rx level
b2..0 = 001 //select DTMF decoder

At this point the user will either go back on-hook or begin to dial a number. Other conditions are ignored.

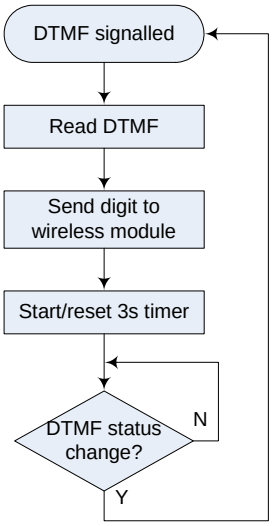
On-hook event signalled by AG1171: SHK = 0

A DTMF dial will be signalled by the CMX865A:
IF interrupt and *Status* ($E6) b5 = 1 then
Write Transmit Mode ($E1):
b3..0 = 0000 //turn off dial tone
Goto DTMF signalled (Figure 4)

A pulse dial will be signalled by the AG1171 as toggling on-hook/off-hook events and must be decoded by the host, so the CMX865A is no longer required.
Write *General Control* ($E0):
b8 = 1 //Enable powersave mode. This sets most registers to default
//including the tx mode register

It may be desirable to provide a long timeout, for example 60 seconds, before signalling an off-hook condition with no other activity. This is helpful in bringing attention to an incorrectly seated handset, or a handset accidentally knocked off-hook. The CMX865A's dual-tone generator can be used to produce a 'howler' tone, a progressively louder dual-tone, which is sent to the handset to help indicate this condition.

Figure 4. On-hook to off-hook detected

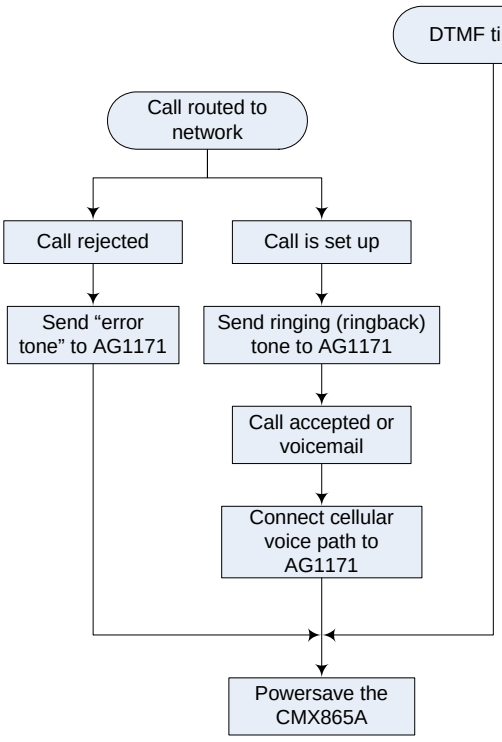


IRQ
Read *Status* ($E6):
b5 = 1 //DTMF detected
b3..0 = *DTMF digit received*

Arbitrary timer to allow for manual (slow) dialling. The length of this delay may be subject to local regulations.

IRQ
Read *Status* ($E6)
b5 = 0 //tone gap detected
b5 = 1 //next tone detected

At this point:
The calling party or the network may be busy or there may be an issue connecting the call.
The dialled number may not be valid.
The call may be set up but not yet answered.
The calling party may be taking too long to dial the call and the 3s timer expires.



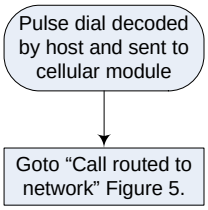
3s timer has expired

All DTMF digits sent to network

The ring tone (ringback) is not sent across GSM so it must be regenerated:
Write *Rx Mode* ($E2),
b15..12 = 0000 //Rx disabled
Write *Tx Mode* ($E1):
//Tx Mode already set for tone transmission
b11..9 = *application dependent* //set Tx //level
b3..0 = 1101 //enable user-defined //tone TB
Write Tx *Mode* ($E1):
b3..0 = 1101 //turn on tone
Delay(n) //delay(n) and delay(m) are used to set the tone cadence
b3..0 = 0000 //turn off tone
Delay(m)

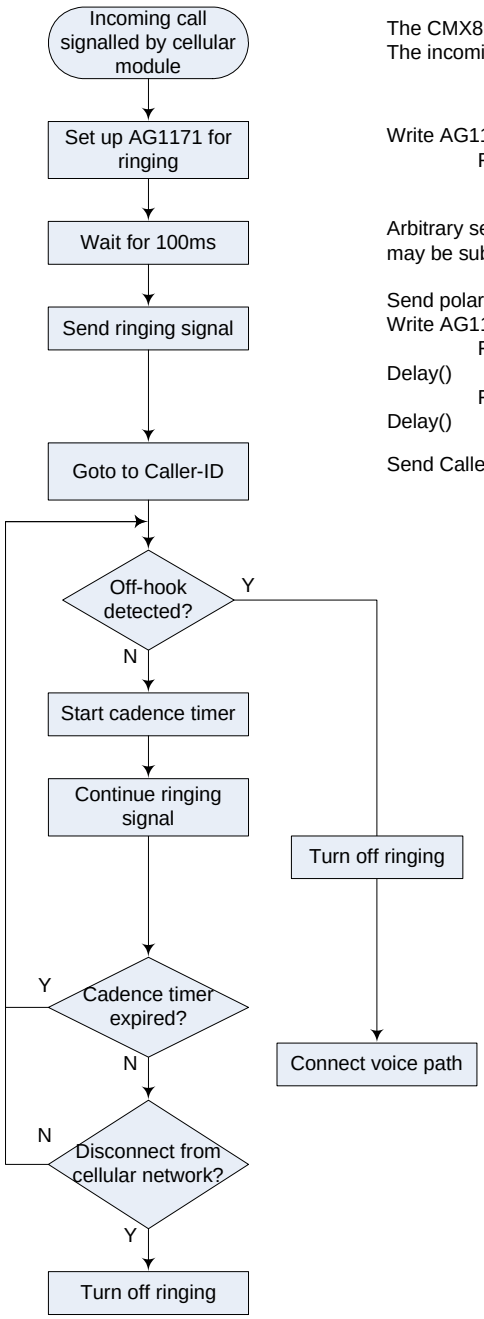
Write *General Control* ($E0):
b8 = 1 //Enable powersave mode. This //sets most registers to default, //including the Transmit mode //register.

Figure 5. DTMF received from POTS line



At this point the cellular network will signal an error if the pulse dial is not valid or
The pulse dial is valid so the cellular module will enable a voice call.
A timeout is recommended to prevent the system blocking if pulse dialled digits are not completed.

Figure 6. Pulse dialling



The CMX865A is not required in this section.
The incoming call will carry a Caller-ID message.

Write AG1171:
RM = 1

Arbitrary settling delay before starting ringing signal. The length of this delay may be subject to local regulations.

Send polarity reversals to the AG1171 to emulate the first ring burst.
Write AG1171:
F/R = 0 //reverse line polarity
Delay() //delay sets ringing signal frequency
F/R = 1 //restore line polarity
Delay()

Send Caller-ID message if required.

During the ringing phase:
Either a POTS phone answers the call
AG1171: SHK = 1 or
the network sends a disconnect. The disconnect will be because the caller has hung-up or because the cellular network has diverted the call – call divert or answerphone.

Cadence timer sets the period between ring bursts

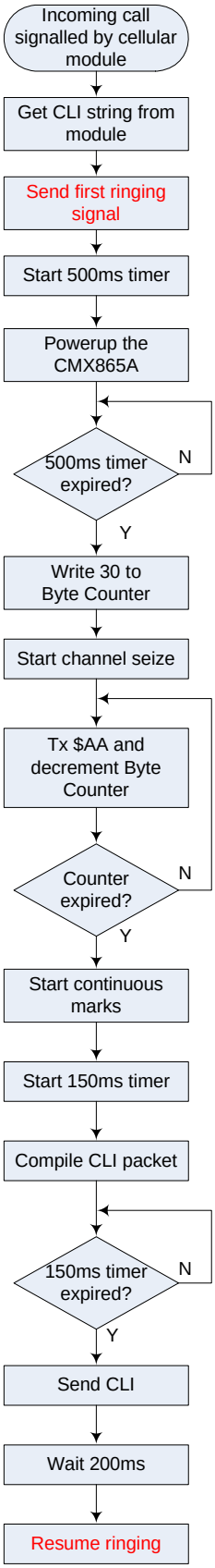
Send next ring burst as above.

An off-hook event is signalled to the host by the
AG1171: SHK = 1
Write AG1171:
RM = 0 //Exit ringing mode
F/R = 0 //ensure normal polarity

Enable cellular audio to AG1171

Write AG1171:
RM = 0 //exit ringing mode
F/R = 0 //ensure normal polarity

Figure 7. Handling an incoming call



A CLI packet can be regenerated from the CLI string received from the cellular module. It can be sent sent over the POTS line as DTMF or FSK following the first ring burst or line reversal depending on the application. Bellcore Type 1 CLI (FSK) is assumed here.

The 'Incoming call' process, Figure 5, is followed until 'send ringing signal' is reached. The following process is then followed.

The minimum silence period before the CLI message is sent is 500ms. During the silence period, the CLI packet can be compiled (reconstructed) using caller-ID string received from the cellular module. See 'CLI packet reconstruction', Figure 7.

Enter Powerup mode
Write *General Control* ($E0) = $0100 //powerup high, TXA and TXAN
//will automatically go to BIAS

Following the silence period is channel seize (1010...) for 250ms. This is to allow the receiving modem to train.
It can be easily generated by sending 10101010 in 8,n,1 asynchronous mode With the start and stop bits added, this yields 0101010101, the required pattern. A byte counter is used to set the length.
1/1200bps x 30 bytes x 10 bits per byte = 250ms

Enable, FSK mode:
Write *Tx Data* ($E3) = $AA //Load 10101010 into tx register
Write *Transmit Mode* ($E1):
b15..12 = 0011 //Bell202 @ 1200bps
b11..9 = *application dependent* //select Tx level
b4, 3 = *10* //asynchronous mode
b2..0 = 110 //8 bits, no parity, 1 stop (8,n,1)
Enable the interrupt if not polling:
Write *General Control* ($E0):
b6 = 1 //Enable IRQN
b3 = 1 //Unmask Tx Data Ready interrupt

On IRQ or poll, Read *Status* ($E6). If b12 = 1 then load the same data byte ($AA) to *Tx Data* ($E3) and decrement the counter until the counter expires.

Following the channel seize is a continuous mark for 250ms. This is to allow the receiving modem's USART to synchronise on the following asynchronous data transfer (the user data). Because the modem is already in asynchronous mode, it will automatically generate continuous mark if the data register is not loaded. Only one interrupt will be generated when *Tx Data* ($E3) is empty and this should be ignored.
Use a timer to set the length of the continuous marks.

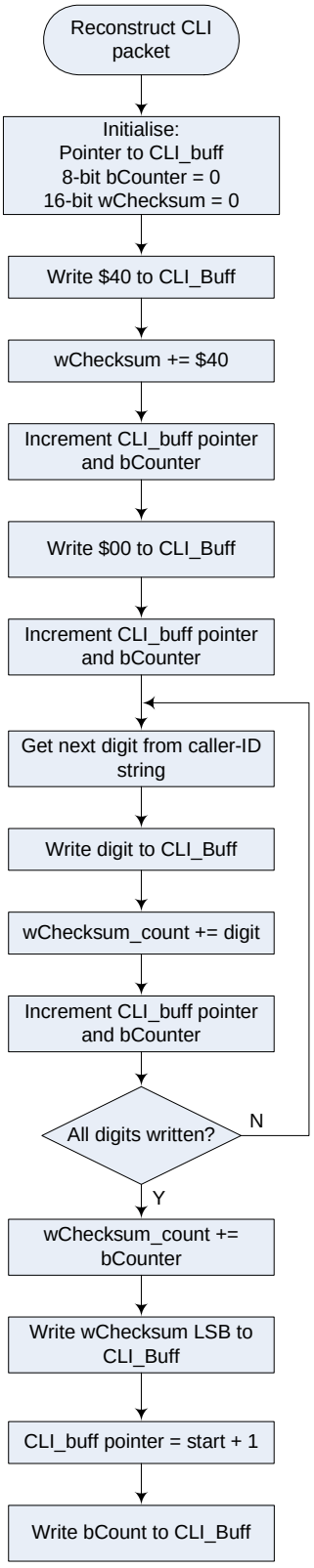
While the continuous mark is being transmitted, the host can compile the CLI packet ready for transmission.

Following the continuous mark, the CLI packet is transmitted. This is also sent asynchronously so the receiving modem's UART will synchronise on the first start bit (1 to 0 transition):
Write *Tx Data* ($E3) = CLI data byte to be transmitted
The interrupt is already enabled if not polling and the write will re-prime the Tx Data Ready flag.
Clear any pending interrupts:
Read *Status* ($E6)

On IRQ, Read *Status* ($E6) or poll if preferred. If b6 = 1 then load next data byte to be transmitted.
Loop until all CLI packet is sent.

Wait 200ms before resuming 'send ringing signal' in 'Incoming call' process (Figure 5).

Figure 8. Generating Bellcore CLI



This process assumes that the CLI string will be fetched from the cellular module in 1 byte pieces, MSB first. I.e. The leading digit of the country code or area code will be the first digit received.
Special messages like “unavailable” or “withheld” can be used in the message format.

Lead byte that identifies the message as SDMF – single data message format.

Checksum calculated on fly.

Point to next location in the message buffer and increment a counter for the number of byte in the packet.

Insert a dummy byte for the ‘message length’ parameter. This is counted on the fly and will be inserted later.

Insert the CLI information one byte at a time.
Calculate the checksum and keep a count of the message length.

All done so add the actual counter value to the checksum.

Append the checksum to the end of the message.

Move the pointer to the 2nd byte in the message buffer.

Replace the dummy byte with the actual message length.

Figure 9. CLI packet reconstruction

6 CLI Structure in Bell Single Data Message Format

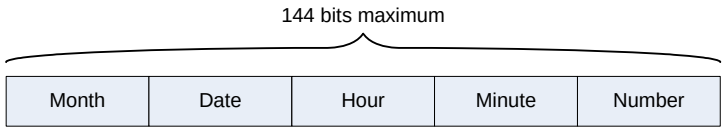


Figure 10. CLI – Calling Parties Number and Time/Date Stamp

Parameter	Bytes in field	Representation	Description
Month	2	January = 01. December = 12	Date of call
Day	2	01 to 31	
Hour	2	0 to 23	Time of call (24 hour)
Minute	2	0 to 59	
Number	10 (max)	-	Number inc. area code

Note that all parameter bytes are ASCII coded

Example:

04 12 31 31 32 24 31 35 35 31 33 33 36 37 34 34 35 30 35 30

04	Message type
12	Length in bytes of message string (18 bytes)
31, 31	Month (November)
32, 24	Date (24th)
31, 35, 35, 31	Time (15:51)
33, 33, 36	Area code (336)
37, 34, 34	Sub area code (744)
35, 30, 35, 30	Local number (5050)

CML Microcircuits (UK) Ltd COMMUNICATION SEMICONDUCTORS	CML Microcircuits (USA) Inc. COMMUNICATION SEMICONDUCTORS	CML Microcircuits (Singapore) Pte Ltd COMMUNICATION SEMICONDUCTORS Singapore	China
Tel: +44 (0)1621 875500 **Fax:** +44 (0)1621 875600 **Sales:** sales@cmlmicro.com **Tech Support:** techsupport@cmlmicro.com	**Tel:** +1 336 744 5050 800 638 5577 **Fax:** +1 336 744 5054 **Sales:** us.sales@cmlmicro.com **Tech Support:** us.techsupport@cmlmicro.com	**Tel:** +65 67450426 **Fax:** +65 67452917 **Sales:** sg.sales@cmlmicro.com **Tech Support:** sg.techsupport@cmlmicro.com	**Tel:** +86 21 6317 4107 +86 21 6317 8916 **Fax:** +86 21 6317 0243 **Sales:** cn.sales@cmlmicro.com.cn **Tech Support:** sg.techsupport@cmlmicro.com

- www.cmlmicro.com -