

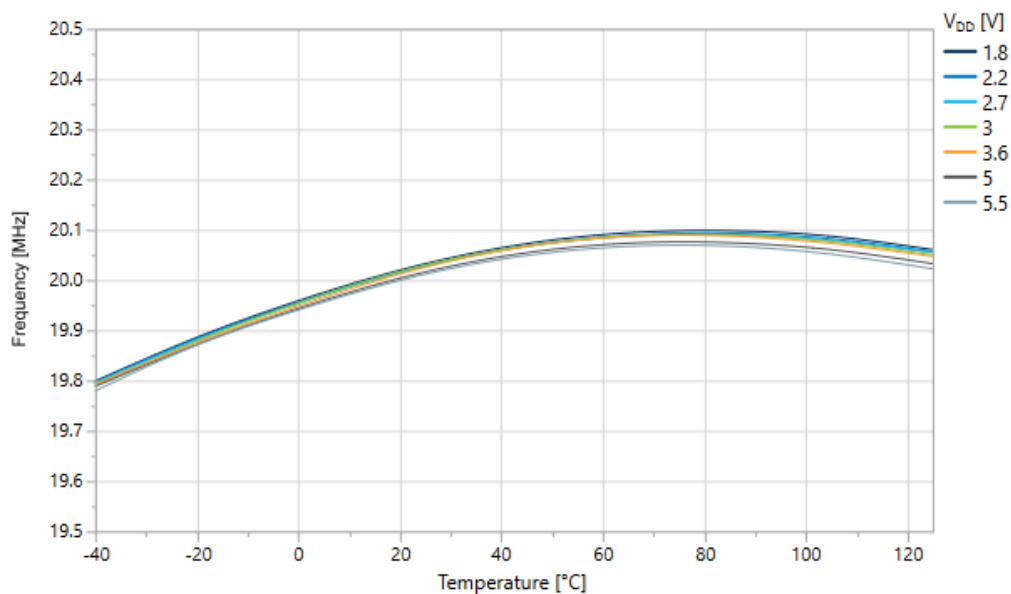
Diskussion angående problemet med CC1310 och WMBUS mottagning av LAN-WMBUS-CX-TH mätare.

Lansen har under ca: 8 månader levererat temperatur- och luftfuktighetssensorn LAN-WMBUS-CX-TH till Kiona (Egain). Lansen har sedan tidigare också levererat mottagarkort kallat OEM-kort.

LAN-WMBUS-CX-TH använder en annan radiosändare än de flesta andra produkterna från Lansen som använder Spirit 2 från ST, då det på grund av leveransproblem då den Spirit 2 kretsen inte gick att få tag. Vår alternativa radiokrets använder en 26 MHz kristall för att få rätt frekvens (868.95 MHz) men data i sig klockas ut från processorn och är alltså beroende av den noggrannhet som processorns egen klocka har.

Attiny1624 och Attiny1626 har följande graf där man ser att hastigheten beror på temperaturen. De första 2 leveranserna.

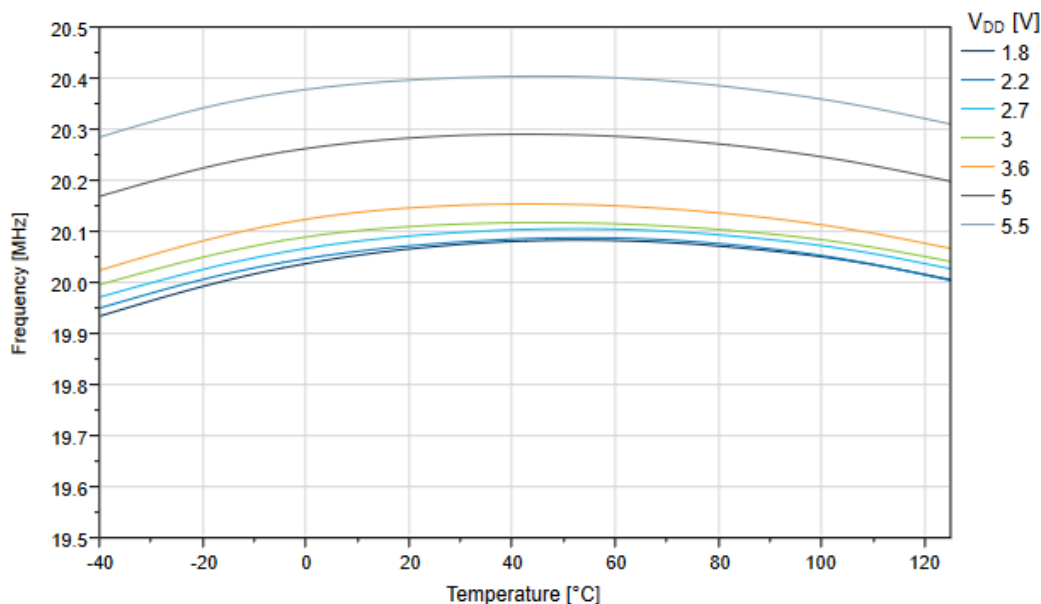
Figure 34-86. OSC20M Internal Oscillator: Frequency vs. Temperature



Tittar man på grafen så ser man att bitfelet från 0 grader till 50 grader är ca: 0,5%

Attiny1616 har grafen nedan, vilken är den MCU som används i den sista 17000 batchen.

Figure 37-72. OSC20M Internal Oscillator: Frequency vs. Temperature



Tittar man på grafen så ser man att bitfelet från 0 grader till 50 grader är ca: 0,75%.

OEM-kortet som ni köpt från oss tidigare och som ni använder i era GW klarar ca: 3% bitfel med den implementationen som är gjord i den.

Alla LAN-WMBUS-CX-TH som vi skickat till er är testade med OEM-kort som acceptanstest, då det är detta kort som vi uppfattat ni använt som mottagarkort.

När data skickas enligt WMBUS så skickas det antingen i S-, C- eller T-mode. De har dock lite olika krav när det gäller bithastighet och bithastighetsfel.

Mode	Maximal bit fel	Bithastighet	Kodning	Deviation	Preamble chips
S	1,5%	32768	Manchester		
T	12%	100.000	3outof6	40-80	48
C	100ppm.	100.000	NRZ	33-56	32

Eftersom T- och C-mode måste kunna tas emot samtidigt så vet man inte på förväg om det är C- eller T-mode som kommer. Utan det vet man inte förrän man tagit emot paketet.

I OEM-kortet har vi vid mottagning valt en "clock recovery algoritm" som klarar ca: 3 % bitfel och blir då nästan kompatibel med S-/T- och C-mode. Det enda den inte klarar av är mätare som skickar T-mode och som ligger väldigt fel, men sådana har vi inte upptäckt finns på marknaden.

LAN-WMBUS-CX-TH skickar data enligt C-mode med kodning NRZ för att hålla nere antalet bytes som skickas för att få så lång batterilivslängd som möjligt. Dock ligger bithastigheten utanför 100ppm som är kravet för att få den godkänd som C-modesändare, dock med goda marginaler inom kravet för T-mode (vi ligger på ca 1% bitfel).

Vi aldrig haft problem att ta emot datat med några då mottagare, då mottagaren inte vet vad det är för data den tar emot. Den har ingen aning om det är T-mode eller C-mode eller vad det är för data över huvud taget.

Vi har testat med Spirit 2, CC1101, en krets från Silabs och Amber wireless sniffer och alla kan ta emot datat an problem.

Mode	Bit rate error in% compared to base rate 100kbit.	Result reception percentage of packet received.	
		Spirit 2	CC1310
C1	2%	100%	0%
C1	0,4%	100%	~10-20%
C1	0,01%	100%	100%
T1	9,2%	0%	~30%
T1	4,8%	100%	100%

Här ovan är mätningarna jag och Markus gjort idag där man ser problemet att CC1310 är väldigt känslig på fel i frekvens.

Följande kopia från CC1200 där den klarar 12% bitfel och är inställbart, samma inställningar finns för övriga kretsar jag någonsin kommit i kontakt med.

6.6 Bit Synchronization

The bit synchronization algorithm extracts the clock from the incoming symbols. The algorithm requires that the expected symbol rate is programmed as described in Section 5.3. Re-synchronization is performed continuously to adjust for any offset between the incoming and programmed symbol rate.

It is possible to select between two different bit synchronization algorithms.

`TOC_CFG.TOC_LIMIT` sets the bit synchronization algorithm and Table 19 below shows the properties of the bit synchronization algorithms.

TOC_LIMIT	Symbol Rate Offset Tolerance	Required Preamble Length
0	< 0.2 %	0.5 byte (only for gain adjustment)
1	< 2 %	2 - 4 bytes
2	Reserved	
3	< 12 %	2 - 4 bytes

Table 19: Bit Synchronization Property

Using the low tolerance setting (`TOC_LIMIT = 0`) greatly reduces system settling times and system power consumption as no preamble bits are needed for bit synchronization or frequency offset compensation (4 bits preamble needed for AGC settling).

6.7 Byte Synchronization, Sync Word Detection

Byte synchronization is achieved by a continuous sync word search using the novel WaveMatch capture logic (correlation filter). The sync word is configured through the `SYNC3/2/1/0` registers and can be programmed to be 11, 16, 18, 24 or 32 bits. This is done through the `SYNC_CFG1.SYNC_MODE` register field. In TX mode, these bits are automatically inserted at the start of the packet by the modulator. The MSB in the sync word is sent first. In RX mode, the demodulator uses the sync word to find the start of the incoming packet.

Nu verkar CC1310 ha en helt annan approach för detta och ti har gjort en speciell firmware som ligger i kretsen och som är helt anpassad till att bara ta emot WMBus data av formatet C- och T-mode eller S-mode. Som jag fattar det är att ti har valt en "clock recovery algorithm" som funkar bra för T-mode men som är helt värdelös för C-mode. Detta för att klara 12% fel i T-mode, men det gör också att den överhuvudtaget inte klarar något fel alls i C mode och det är därför det är så många av LAN-WMBUS-CX-TH som inte kommer in.

Jag har sökt Texas och försökt få reda på vilken inställning som man kan ändra på för att få enheten mer okänslig för bithastighet onoggrannhet och för att den ska fungera med ett större bitfel i C-mode.

Jag har också frågat om man kan köra kretsen utan WMBUS parsern inkluderat så som vi gör i "gamla" OEM-kortet så man kan minska känsligheten för bitfel.

Här är lite andra diskussioner där samma typ av ärende diskuteras och där det finns ändringar man kan göra.

<https://e2e.ti.com/support/wireless-connectivity/sub-1-ghz-group/sub-1-ghz/f/sub-1-ghz-forum/631901/cc1310-data-rate-offset-tolerance>

Part Number: CC1310
Other Parts Discussed in Thread: CC110L,

Hello,

we are trying to integrate a device using the CC1310 transceiver into an existing ecosystem of devices using different transceivers. Most of these devices use a data rate of 20.4kbaud, but some use 20.8kbaud. Compatibility has not been an issue before, since the transceivers used offered the ability to compensate for data rate offset. For example: one of the transceivers used in our system is the CC110L, it offers programmable data rate offset tolerance of up to 3.125% (BS_LIMIT[1:0]).

The CC1310 however seems to have a fixed data rate offset tolerance of typically 1600ppm. Is there an override or a patch to change this? We require higher tolerance to make the CC1310 work in our system. We can not change the data rate of other devices since they are already deployed.

Best regards

over 6 years ago

Like More



● Sverre over 6 years ago

🔥 TL_Mastermind 39860 points

2% data rate offset is challenging, if not impossible, without a patch. We do not have a generic patch that supports 2% data rate offset. We have patches that support specific protocols with up to 12% data rate offset tolerance, but they are not applicable in this case.

There is a register that can be tested. Address 0x4004 50C8. We have not performed any detailed testing with changes to this register.

[9:7]: Symbol timing error accumulator period (from 4 to 128 symbols)

[6:4]: Symbol timing error accumulator gain (from 1/512 to 1/4)

Set gain to max (0x7). Start with max period (=0x5) and test with all settings down to 0. Use long packet lengths.

The register can be changed through a HW_REG_OVERRIDE, but make sure to keep all other bit fields than [9:4] unchanged.

A patch could be an option, but only if the volumes justify the development. Please drop me a PM if you want to discuss this privately.

^ 0 v More

Här kanske det är ngt av ändringarna som man kan göra för att få till det här.

There is a register that can be tested. Address 0x4004 50C8. We have not performed any detailed testing with changes to this register.

[9:7]: Symbol timing error accumulator period (from 4 to 128 symbols)

[6:4]: Symbol timing error accumulator gain (from 1/512 to 1/4)