

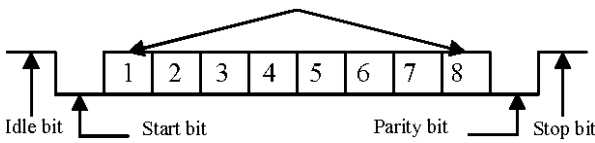


Content:

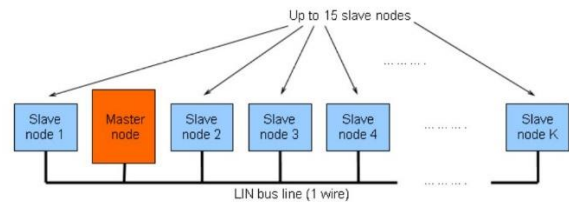
- 1- lin introduction
 - a. local inter-connect network
 - b. low cost low speed
 - c. UART based
 - d. Single wire
 - e. Single master multi slave
 - f. Broad cast
 - g. 16 node
 - h. Self-synchronization
 - i. Message ID
 - j. Collision detection
- 2- lin operation concepts
 - a. master task (header)
 - b. slave task (response)
- 3- lin frame structure
 - a. header
 - i. sync break
 - ii. sync byte
 - iii. message ID
 1. data length
 2. message priority
 3. parity bits
 - b. response
 - i. data
 - ii. check sum
 - c. inter byte space
 - d. response space
- 4- lin communication type
 - a. data from master to slave
 - b. data from slave to master
 - c. data from slave to slave
- 5- lin frame type
 - a. unconditional frame
 - b. event triggered frame
 - c. sporadic frame
 - d. diagnostic frames
- 6- lin bus timing and scheduling table
 - a. lin is time triggered protocol
 - b. calculate frame time
- 7- lin error handling
 - a. parity bits
 - b. check sum
 - c. application solution
- 8- lin sleep and wake up
 - a. sleep
 - b. wake up
- 9- lin connections in automotive ECU

1- LIN introduction

- اختصار local inter-connect network
- عملاء VOLVO وهدفه انه يقي low cost , low speed عشان اوصله مع ال sensors و ال actuators و ال sensors اصلا بتبع data قليله فمحتاج ابعث ال data ب rate كبير
- ال license بتاعته free
- ال Lin هوا UART based protocol (يعني لما بيعت اى byte بيعتها بشكل ال UART يعني بيعت start bit وبعديها ال byte و stop bit) Data bits

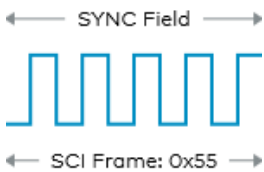


- ال lin هوا single wire bus
- Single master multi slave



- ليه single master ؟ ...
- عشان هوا بي communicate with sensors , actuators و دول مش منطقي انهم يبقو master دايما ال ECU هيا ال master
- شغال broad cast يعني ممكن بيعت message و كل ال slaves يستقبلوها
- عدد ال nodes (node is a device connected on bus) ... 16 node
- Max speed 20 K bit/sec
- ليه مكان ال UART ؟
- عشان مشاكل ال UART ذى

- o ال UART كان بيتكلم مع device واحد
- o مفيش error checking فى ال UART غير ال parity bit (ال lin بيستخدم ال check sum error checking)
- o اخره بيعت 1 byte ... (انما ال lin يقدر بيعت 1 or 2 or 8 byte)
- ال lin عنده self-synchronization يعني ايه ؟

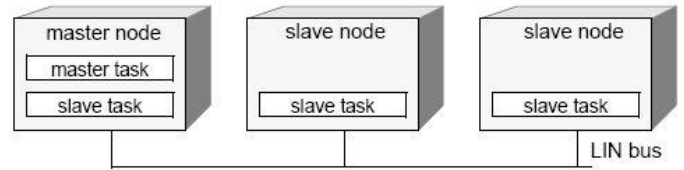


- يعني بيضبط ال clock الى هيتكلم بيها مع باقى الناس ... ازاي ؟
- بيعت byte فى اول كل frame عبارة عن 01010101 ... 0x55
- قالى يستقبلها يقدر منها يفهم ال master هيتكلم بانتهى ال speed ال byte دى اسمها synchronization field بالتالى انا مش محتاج ابعث clock بالتالى هوا single wire
- ال LIN بيعتمد على ال message ID ... يعني ال master بينزل على ال bus ... message ب address معين ... الى مهمت من ال slaves يقرأ ال message دى
- يقرأها و الى مش مهمت ... عنه ما قراها
- فانا بظبط ال configuration بتاعت ال slaves انهم يقرأو ال message دى (و دا بيقلل ال bandwidth لاننى مش هبعث ال message اكثر من مره دى ما كنت بعمل فى ال I2C , SPI)
- مفيش collision detection ... لان دايما ال master هوا الى هيبعت ... فعمر ما هتيجى case ان اتنين device يتكلمو فى نفس الوقت

- افكر ان ال
- 0 -> dominant bit
- 1 -> recessive bit
- و دايما فى اى communication protocol ال idle = 1
- عشان لما يجى حد يتكلم بيعت 0 فال 0 هوا الى يظهر على ال bus

2- Operation concept

ال LIN frame لما بيجي بتبعته ... مش بيبعته حد واحد ... بيتشارك فى تكوينه 2 tasks ...
يعنى ايه ؟



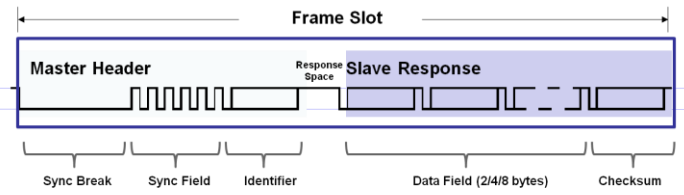
كل node فيها حاجة اسمها tasks ... يعني مثلا
ال master node فيها 2 tasks

- Master task
- Slave task

و ال slave node عندها

- بس Slave task

و عشان ال frame يتكون ... بيتقسم لجزئين ... واحد بيكونه ال master task و اسمه master header و التاني بيكونه ال slave task سواء الى فى ال master او ال slaves و اسمه slave response



From MASTER to SLAVES

From addressed SLAVE to MASTER and all other SLAVES

دايما اول جزء فى ال frame ال master هوا الى بيكونه

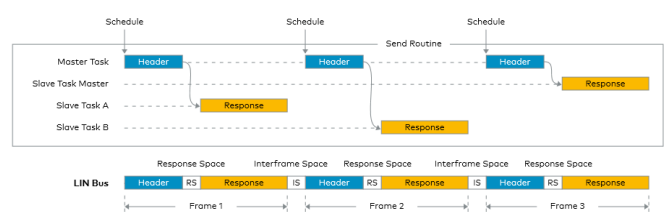
1- Master task

- دايما موجود فى ال master node
- دايما اول واحد بيشتغل ... لانه اول واحد هيبعت
- جواه 3 حاجات
- o Sync break
- o Synch field
- o Identifier (message ID)
- ال master task بيتحكم فى ...
- o ال bus عليه message ولا لا
- o بيتحكم فى ال communication لانه بيعت اول نص ال frame
- o بيحدد ال message priority ... لانها بتعتمد على ال message ID
- o بيحدد سرعه ال transmission لانه بيعت ال sync byte
- o بيستقبل ال wake up requests الى جايه من ال slaves

2- Slave task

- موجوده جوا ال master node و جوا كل slave node
- مسؤله عن تكوين النص التاني من ال frame الى اسمه response ال response بيتكون من ال
- o Data
- o Check sum

Message scenario :



- 1- ال master بينزل على ال bus ال header ... الى جواه ال message ID
- 2- ال header لما ينزل يروح لكل slave task على ال bus سواء كانت slave task جوا ال slave node او حتى جوا ال master node (هيا بتروح لكل slave task على ال bus)
- 3- على حسب ال message ID كل slave task معمولها configuration انها تعمل حاجه من 3
- a. Slave task -> send data
- b. Slave task -> receive data
- c. Slave task -> do nothing
- مثلا اول frame (من ناحيه الشمال) الى كونه ال master task فى ال master node و ال slave task الى فى ال slave node A (slave task A) كان معمولها

configuration انها لما تشوف ID message دا (الى اتبعت في ال header) ...
ترد هيا بال response

- ثاني frame كان ID مختلف و ال slave task الى في node B هيا الى كانت المفروض تطلع ال response
- ثالث frame الى طلع ال header هوا ال master task in master node و الى طلع ال response هوا ال slave task في ال master node

3- lin frame structure

a. header

i. sync break

- هدفه : ان ال master بيصحى الناس الى على ال bus عشان هيبت header
- عبارته عن 13 bit = 0 including the start bit
- اخرها 1 bit = 1 break delimiter
- ليه 13 bit ؟ عشان ال slaves ياخذو وقتهم على ما يصحو

ii. sync byte

- هدفها : تضبط ال speed
- عبارته عن 0x55 و اولها start , stop برضو

iii. message ID

- هدفه : عشان كل slave task تعريف ال action الى هتعمله based على ال ID دا
- سواء هتبع او هتستقبل او مش هتعمل حاجة ... بيسموه protective ID
- هوا 8 bit ... بس ال ID فيهم 6 bit بس ... يعنى اخرى بيقى عندي 64 message
 - o من 0-59 دول messages عاديه بتشيل data
 - o 60 , 61 اسمهم diagnostic frames
 - o 62 , 63 for future use
- ال message ID بيحددلى ...

o Frame length

ID range	Frame length
0-31 0x00 - 0x1f	2
32-47 0x20 - 0x2f	4
48-63 0x30 - 0x3f	8

- o Priority
- o Parity bits

ال 2bits الى باقين دول parity bits

- o اول parity bit عبارته عن XOR(0,2,4)
- o ثاني parity bit عبارته عن XOR(1,3,5)

b. Response

i. Data

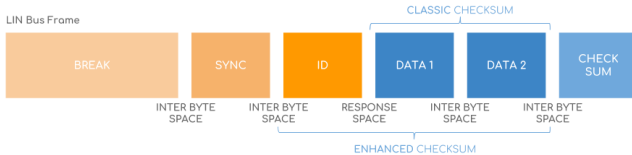
- بتكون من 2 or 4 or 8 bytes على حسب ال message ID
- كل byte فيهم اولها start bit و اخرها stop bit ال UART

ii. Check sum

- يجمع ال data و هوا بيبيعت بيبيعت ال summation
- ال receiver يستقبل ال data و بيعت ال check sum و يقارنه بال check sum الى جاله
- ال check sum الى ال LIN شغال بيه اسمه CRC(cyclic check sum)
- الى بيبيعت ال inverted check sum
- ال receiver بيحسب ال check sum و يجمعه مع ال inverted لو طلع 0xFF بيقى كذا ال data جايه صح

c. Inter byte space

دى وقت بيحط بين ال bytes عشان لو حصل شويه delay او عشان يلحق ال sensors تعمل process على ال byte لان برضو ال sensors هيا low cost فال accuracy بتاعتها مش كويسه فمكن مثلا ال sensor بتاخر 3 or 4 bits مثلا فانا ممكن اسمح بتاخير بين كل byte و الثانيه

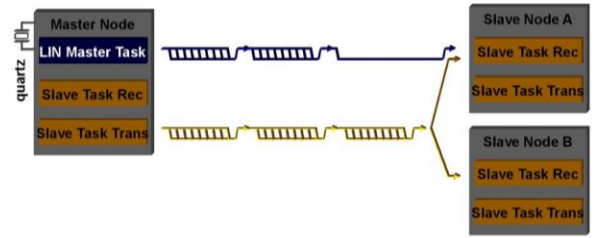


d. Response space

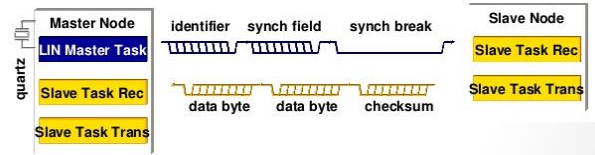
- دا delay بين ال header و ال response
- محتاجه ليه ؟ ... لان لما ال master ينزل header كل slave task المفروض تقرأ ال message ID و تقرر على اساسه هل هتبع ال response ولا هتستنى response من حد ثاني ولا مش هتعمل حاجة
- عشان تاخذ القرار دا بتاخذ وقت ... فلانم بيقى في delay بين ال header و ال response على ما ال slave tasks تقرر
- الوقت دا بيقول قليل جدا لو ال slave task الى عند ال master node هيا الى هتبع ال response

4- lin communication type

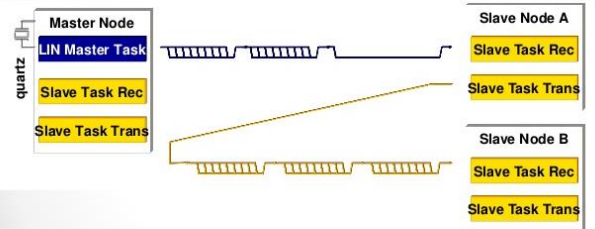
a. data from master to slave



b. data from slave to master



c. data from slave to slave



و في كل الحالات ال master هوا ال بيبدأ ال communication باله بيعت ال header

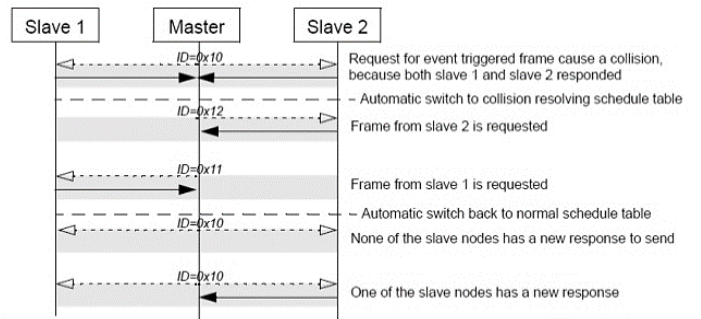
5- lin frames types

a. unconditional frames (data frames)

بتشيل message ID 0-59

b. Event trigger frame

- بتشيل برضو من 0-59
- هدفه : يقلل استخدام ال bus ... ازاي ؟
- انى هخلى ال slave ميردش على ال master غير لما يكون في عنده update في ال data ... مثلا قرايه السينسور اتغيرت ... ساعتها يرد ... انما طول ما هيا ذى ما هيا هيرد ليه
- بالتالى اوfer ال bandwidth



مثلا ال master بيعت header عشان يعرف حاله الابواب ... ف الباب الى عنده

- update بس هوا الى هيرد
- طب لو الاتنين ردو في نفس اللحظة ... ال master هيعتبره collision فيبتدى يكلم واحد واحد ب unconditional frame و كل واحد كلمه ب ID لوحده
- كل ما كبر ال ID كل ما ال priority بتبقى اعلى

c. Sporadic frames

- ذى ال event trigger بس لو ال master هوا الى هيبعت ال frame كله
- يعنى ال master هيبعت ال header ... بس امتى هينزل ال response لو عنده update لو معندوش مش هيعمل حاجة ... فياقلى ال slaves تفهم ان طالما ال master مبعثش response جديد بيقى هيا نفس ال data القديمه
- و برضو هدفه يقلل ال band width

d. Diagnostic frames

- دى الى هيا ال commands ذى ال wake up command or sleep command

6- lin bus timing and scheduling table

LIN is time triggered protocol

- ال LIN هو time triggered protocol يعني كل frame ليها وقت معين تنزل فيه على ال bus
- مش ذي باقي ال protocols وقت ما احب ابعث هبعث
- بالتالي هوا عنده schedule table فيه وقت كل header و بالتالي لازم ابقى عارف كل وقت ال LIN frame و قته قد ايه عشان اعرف هبعث ال frame الى بعده امتي

Calculate frame time

Header =

sync break (13 bit) + delimiter (1 bit) +
sync byte with start , stop(10bit) + ID with start , stop (10bit)
= 34 bit

Response = data ((2 or 4 or 8 byte) + check sum (1byte)) * 10

$$t_{frame} = \text{bit number} * \frac{1}{\text{bit rate}}$$

Example : For 8-byte data

Bit number = 34 + (8 + 1) * 10 = 124 bit

$$t_{frame} = 124 * \frac{1}{20 \text{ kbit/sec}}$$

و نضيف ال response space , interbyte space , factor 40% الى هيا 1.4

$$t_{frame} = 124 * \frac{1}{20 \text{ kbit/sec}} * 1.4$$

و في العربية ال header هينزل مثلا على ال bus كل 10 ms مثلا ... عشان ا check ال ابواب ...
على حسب نوع ال frame ال ابواب بقي هتشوف

- هتزد على طول
- لو في update هتزد ... لو مفيش مش هتعبير ال master

7- lin error handling

- ال error handling مش جزء من ال LIN ... يعني لو حصل error هيقوللى ان في error بس عن طريق FLAG بس مش هيعمل action
- ايه ال errors الى ال LIN بي check عليها ؟

o Parity bits (2 bit in header) check error in ID

o Check sum (1byte in response) check for data , ID

- ال receiver هوا الى بيشفوف ال هوا الى بيشفوف ال errors دى ... بس ال master ميعرفش ان حصل error

الحل ؟

انى اعمل message ب ID معين ... لو بعثته ال slave يرد عليه بقيمة ال flag error bits الى عنده و بناء عليه ال master يشوف هيعمل ايه

8- lin sleep and wake up

sleep

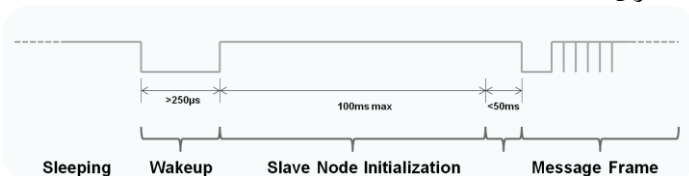
- ممكن ال LIN bus كله بكل ال nodes الى عليه ادخلهم في sleep mode
- هيشخ في ال sleep mode امتي ؟
 - o لو ال bus فضل idle اربع ثواني
 - o لو ال master بعث diagnostic frame (sleep command) with ID = 60
- ال data فيها 8 byte اول 0 byte و الباقي كله 0xFF

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x00	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF

الهدف : عشان يقلل ال power consumption

Wake up

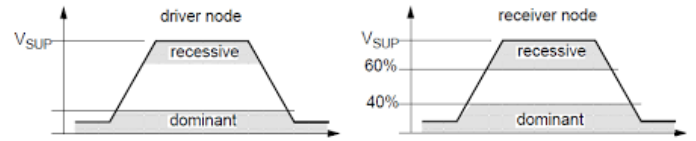
- المسؤول عن ال wake up هوا ال slave task لانها بتبقى عايزة تبعت data في وقت معين مثلا بس لازم ال تصحى ال master task عشان بيعتليها ال header
- ازاي ؟



- 1- ال slave بيعت 0 من 250 micro و يستنى لغايه ما ال master يقوم و يشغل ال scheduler و يجي الدور على ال message الى ال slave دا مستنيها فيرد عليه بالداتا الى عايز بيعتها
- 2- لو ال master مقامش ابعثله request تاني و تالت
- 3- لو مقامش معد الرابع مينفعش ابعث request رابع غير بعد 1.5 sec

9- lin connections in automotive ECU

ال voltage بتاع ال LIN bus هو 12 volt



شغال ازاي ؟

- في ال receiving

- o 1 > 60% of supply voltage
- o 0 < 40%

عشان دايما في ال receiving بيبقى في noise فينبيله margin اكبر
- في ال transmit

- o 1 > 80%
- o 0 < 20%

عشان احول من logic ال logic ل controller (5v) ال LIN Bus لازم استخدم transceiver
دايما ال transceiver هوا external hardware

