

BEÁGYAZOTT TELEGRAMOKAT TARTALMAZÓ IPARI ETHERNET KERETEK CIKLUSIDEJE

CYCLE TIME OF EMBEDDED DATAGRAM USED INDUSTRIAL ETHERNET FRAMES

Ferenczi István*

ABSTRACT

There are two ways to forward control data in Hard Real-Time industrial Ethernet communication systems. The first approach is that the control device (master) sends separate frames containing the control data to IO devices (slave) that are connected to it via Ethernet. The second approach, one tries to utilize the maximal payload that amounts to 1500 bytes according to the IEEE 802.3 standard. In this case, one attempts to place the control data (telegrams) so as to fully utilize frames. The EtherCAT system developed by Backhoff is based on this principle, which is more advantageous when using small sized (16-64 bytes) telegrams.

1. BEVEZETÉS

Az elsősorban motorvezérléseknél használt, szigorúan valós idejű ipari Ethernet kommunikációs rendszereknél a vezérlési adatok továbbítása két módon valósítható meg. Az egyik megoldás, amikor a kontroller mindegyik, vele Ethernet kommunikációs kapcsolatban lévő IO eszköznek külön-külön címezi a kereteket, amelyek a megfelelő vezérlési adatokat tartalmazzák. Ezt a módszert alkalmazza a Profinet IRT rendszer. Ebben az esetben 36 bájt nál kisebb vezérlési adatok gyakorlatilag nem befolyásolják a továbbításhoz szükséges időt, mert így a teljes keretméret nem haladja meg a minimális, 64 bájt os keretet. Hátránya ennek a megoldásnak, hogy az előbb említett esetben a keretterhelési tényező a legjobb esetben is csak 56% lehet. Igazi előnye az 1000Mbps-os gigabit Ethernet hálózaton mutatkozik meg, amikor az adattovábbításhoz szükséges idő gyakorlatilag már nem függ a keret méretétől. [1]

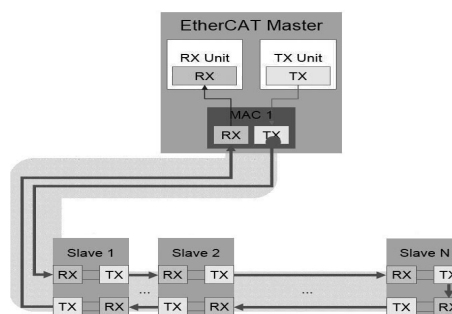
A másik megoldás, amikor arra törekszünk, hogy kihasználjuk a maximális keretterhelést, amely az IEEE 802.3 szerint 1500 bájt lehet. Ebben az esetben az IO eszközök számára továbbítandó vezérlési adatokat (telegramokat) igyekszünk egy vagy több keretben belül úgy elhelyezni, hogy minél nagyobb legyen a keretek kihasználtsága. Ez a módszer kétségtelenül előnyt jelent kisméretű (36 bájt nál kisebb) telegramok továbbítása esetén.

* főiskolai adjunktus, Nyíregyházi Főiskola, MMK

Ebben a cikkben egy ilyen megoldást és egy ehhez tartozó ciklusidő számítási algoritmust fogok bemutatni.

2. AZ EtherCAT RENDSZER FONTOSABB JELLEMZŐI

Az EtherCAT-et igen rövid ciklusidő, zavarmentes kommunikáció és nagy pontosságú szinkronizálás jellemzi. 100 Mbps-os full duplex Ethernet hálózaton, szegmensenként gyakorlatilag szinte korlátlan számú (max. 65 535) IO eszköz kapcsolható össze egymással. A gyártó adatai szerint 100 IO eszköz, összesen 1000 digitális IO csatorna esetében a frissítési idő mindösszesen 30 μ s, de 100 darab szervo tengely, egyenként 8 bájt os IO adatokkal történő hajtásakor sem sokkal haladja meg a 100 μ s-ot [3]. E rendkívüli gyorsaságot annak köszönheti, hogy a csomópontokban lévő IO eszközök „menet közben”, közvetlen memória hozzáféréssel (DMA) veszik le a nekik címzett, illetve teszik fel a továbbítani kívánt adatokat (1. ábra). Ennek következtében a csomópontokon belüli késleltetési idő is rendkívül rövid, mindösszesen 1,35 μ s full duplex hálózaton.



1. ábra. EtherCAT adattovábbítás

A vezérlési adatok nagysága sem mondható korlátozottnak. Noha a motorvezérléseknél nincs szükség nagyméretű telegramok továbbítására, a gyártó 60 kilobájtban határozta meg a maximális adatmennyiséget, amely továbbítható a csomópontoknak. (Ebben az esetben természetesen több keretre van szükség.)

A rendszer elsősorban a busz topológiát támogatja, de jól használható, csillag, fa és vegyes topológiájú hálózatokban is. Vegyes topológiában a ciklusidő azért lényegesen nagyobb, mint például a busz topológiában.

3. AZ EtherCAT ESZKÖZ-PROTOKOLL

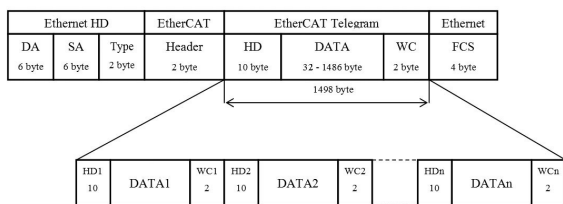
Egy teljes EtherCAT rendszer alapvetően két féle kommunikációs protokollt használ, amelyek a szabványos IEEE 802.3 Ethernetre épülnek:

- Nem szinkronizált folyamatokhoz, mint például konfigurációs beállításokhoz, folyamatvizualizáláshoz valamint lekérdezésekhez a TCP/IP vagy az UDP/IP-t,
- Szinkronizált vezérlési adatok továbbítására az EtherCAT eszköz-protokollt.

Ebben a cikkben csak az utóbbi kerül bemutatásra.

Ahhoz, hogy a busz ciklusidő meghatározásához szükséges algoritmust felépíthessük, vizsgáljuk meg először, hogy hogyan épülnek be a telegramok az Ethernet keretbe. Az IEEE 802.3 szabvány szerinti Ethernet keret annyiban módosul, hogy kiegészül egy 2 bájtos EtherCAT fejléccel, amely főleg a beágyazott adatmennyiség hosszára és típusára vonatkozóan tartalmaz információt. Így az adatok számára egy kereten belül már csak 1498 bájtnyi hely marad, ahová az EtherCAT telegramokat helyezhetjük. A keret azonosítására az ipari Ethernet rendszereknél használatos Ether Type-al történik, amely ebben az esetben 0x88A4. [2]

Azonosítás céljából mindegyik beágyazott telegramot is fejléccel (Telegram Header), illetve ellenőrző résszel, ún. működés számlálóval (Working Counter) kell ellátni (2. ábra).



2. ábra. Ethernet keretbe ágyazott telegramok

A keret maximális mérete 1518 bájt, amely még kiegészül 8 bájtnyi előtaggal (7 bájt Preambulum + 1 bájt SFD), valamint 12 bájt keretek közötti réssel (IFG), így a teljes keretméret, összesen 1538 bájt lesz.

4. A CIKLUSIDŐ MEGHATÁROZÁSA

A busz ciklusidő meghatározása meglehetősen összetett feladat, de alapvetően két tényezőtől függ: a csomópontok számától (N) és a keretterheléstől ($Data$). Először – a keretterhelés függvényében – meg

kell határoznunk, hogy hány telegram helyezhető el egy keretben. Figyelembe véve, hogy mindegyik telegramhoz tartozik egy 10 bájtos fejléc (HD) és egy 2 bájtos ellenőrző rész (WC), következik, hogy egy kereten belül elhelyezhető telegramok számát (k) a következő összefüggéssel határozhatjuk meg:

$$k = INT\left[\frac{Max_eCAT_Data_Size}{HD + Data + WC}\right] = INT\left[\frac{1498}{12 + Data}\right] \quad (1)$$

ahol $INT[]$ a tört egész részét jelenti.

A továbbiakban bemutatott modellben feltételezzük, hogy egyszerre, egy időben ugyanakkora adatmennyiséget továbbítunk minden csomópontbeli eszköznek. Ez nem egy rendkívüli dolog, mivel a valóságban is, például tengelyvezérléseknél, ha az eszközök azonosak, akkor a vezérlésükhöz szükséges adatmennyiség is azonos nagyságú, csak a tartalmuk más.

Ha már ismerjük az egy keretben elhelyezhető telegramok számát (k) és ez a szám kisebb, mint az eszközök száma, akkor meghatározhatjuk, hogy a teljes adatmennyiség továbbításához hány keretre (n_F) van szükség.

$$n_F = INT\left[\frac{N}{k}\right] + 1; \quad \text{ha } N > k \quad (2)$$

Ellenkező esetben a teljes adatmennyiség elfér egyetlen egy keretben és k egyenlő lesz N -el. (k a telegramok, N pedig a csomópontok száma.)

Figyelembe véve az előbbi kritériumokat és azt a tényt, hogy a minimális Ethernet keretméret nem lehet kisebb, mint 64 bájt, 3 esetet különböztethetünk meg:

- 1) $N = k$ és a teljes EtherCAT telegramok mérete (fejléccel és működés számlálóval együtt) nem haladja meg a 44 bájtot. Ez a legegyszerűbb eset, ilyenkor a teljes keretméret 64 bájt, az előtaggal pedig 84 bájt lesz. A ciklusidő (T_c) meghatározásához figyelembe kell vegyünk még a csomóponti eszközök, valamint a közeg késleltetési idejét ($t_D = 1,35\mu s$ és $t_M \approx 0,5ns/m$; maximum 100m-es szegmenseknél $t_M = 0,5\mu s$). Következik:

$$T_c = \frac{8 \cdot 84}{b} + N \cdot (t_D + t_M) \quad (3)$$

Az előbbi összefüggésben b a bitebességet jelöli megabit/szekundumban (Mbps), a ciklusidőt pedig mikroszekundumban kapjuk.

Megjegyzés: A fenti eset a valóságban gyakorlatilag nem, vagy csak nagyon ritkán fordul elő, ugyanis például 3 csomópont és 1 bájt vezérlési adat esetében a keretterhelés a fejlécekkel együtt már meghaladja a 34 bájtot!

- 2) $N = k$ és a teljes vezérlési adatmennyiség elfér egyetlen egy keretben ($n_F = 1$). Ebben az esetben is figyelembe kell, vegyünk az EtherCAT telegra-

mok mellett az Ethernet fejléccet és egyéb járulékos bájtozat. A keret nagysága változó lehet, de nem haladja meg a maximális méretet. A ciklusidő a következő összefüggéssel számítható ki:

$$T_c = \frac{8N \cdot (12 + Data) + 320}{b} + N(t_D + t_M) \quad (4)$$

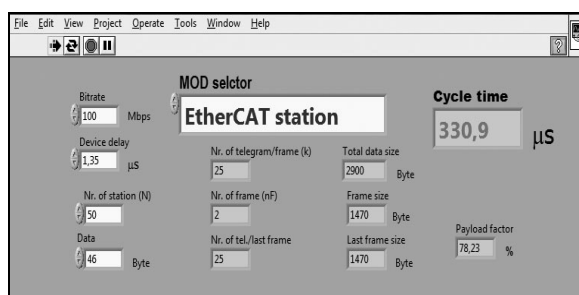
3) A legáltalánosabb eset az, amikor a csomópontok száma nagyobb, mint ahány telegram elhelyezhető egy keretben ($N > k$). A (2) alapján kapjuk meg, hogy hány keretre van szükség a teljes vezérlési adatmennyiség továbbítására. A ciklusidőt pedig az alábbi általánosított összefüggéssel határozhatjuk meg:

$$T_c = \frac{8}{b} [40 \cdot n_F + N \cdot (12 + Data)] + N \cdot (t_D + t_M) \quad (5)$$

Megjegyzés: A (3), (4), (5) összefüggésekben szereplő t_D és t_M , különböző értékeket kaphat. A késleltetési idő (t_D) függ a bitsebességtől, 1Gbps-es hálózatban 0,85 μ s, t_M pedig az eszközök közötti távolságtól.

5. A SZÁMÍTÁSI ALGORITMUS KIDOLGOZÁSA LABVIEW 8.2-BEN

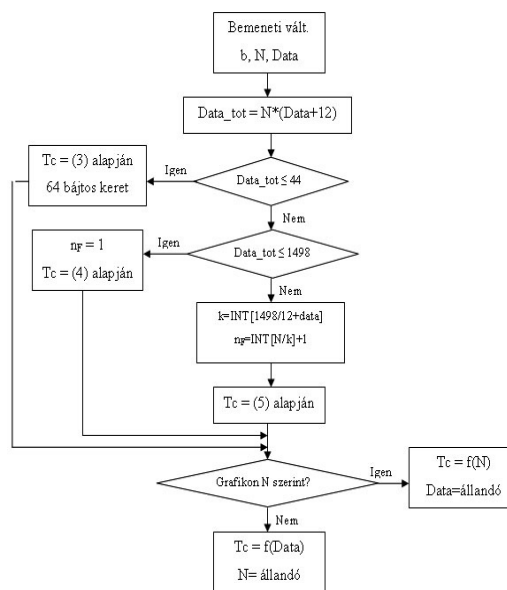
A Labviewban kidolgozott algoritmus mindhárom, az előző fejezetben bemutatott esetet kezelni tudja és minden esetben lehetőséget biztosít a ciklusidő meghatározására gyakorlatilag korlátlan számú csomóponti eszköz (N) és 1-től egészen 1486 bájtig terjedő keretterhelés függvényében. Egyaránt alkalmazható Fast Ethernet és Gigabit Ethernet hálózatra is, ennek megfelelően beállítható az eszközkésleltetési idő is. Az eszközök közötti szegmenshossz minden esetben a maximális 100 méter hosszúságú.



3. ábra. Az alkalmazás konfigurációs felülete

A ciklusidő kiszámítása mellett, ábrázolni tudja ennek változását a csomópontok, illetve a keretterhelés függvényében, a keletkező adatokat pedig Excel fájlban is tudja tárolni. Ezen kívül, mint ahogyan a 3. ábrán is láthatjuk, számos más információt is kijelöl, mint például a keretek, illetve a beágyazott telegramok számát, vagy a keretterhelési tényezőt [1].

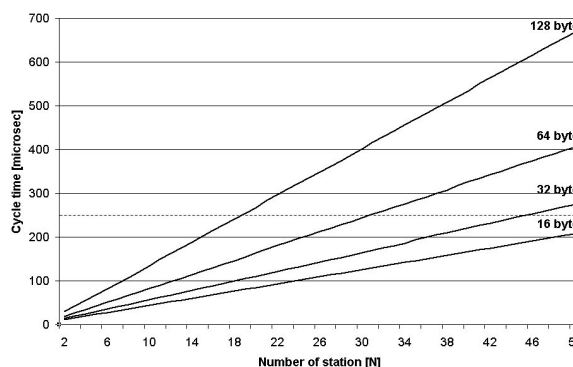
A következő ábrán a Labviewban elkészített EtherCAT minimális ciklusidő meghatározási modelljének algoritmusát láthatjuk.



4. ábra. Az EtherCAT ciklusidő számítási algoritmus

6. A CIKLUSIDŐ VÁLTOZÁSÁNAK ELEMZÉSE

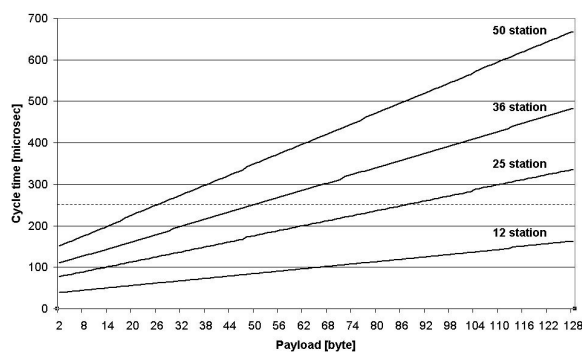
Először vizsgáljuk meg azt az esetet, amikor a ciklusidő a csomóponti eszközök száma szerint változik, maximum 50 csomópontig különböző, 16, 32, 64 illetve 128 bájttal keretterhelés mellett, Fast Ethernet (100 Mbps) hálózatban (5. ábra).



5. ábra. A ciklusidő változása az IO eszközök száma szerint

A szigorúan valós idejű rendszereknél a ciklusidő legfeljebb 1ms, vagy ennek törtrésze 1/2, 1/4, 1/8, stb. lehet. 50 csomópont esetében 16 bájttal eszköznkénti keretterheléssel még a 0,25ms-os ciklusidő is megvalósítható. Nagyobb keretterheléseknél, például 64 bájtnál már csak 0,5ms-os ciklusidő biztosítható.

A 6. ábrán ugyancsak a ciklusidő alakulását láthatjuk, csak most a keretterhelés függvényében.



6. ábra. A ciklusidő változása a keretterhelés szerint

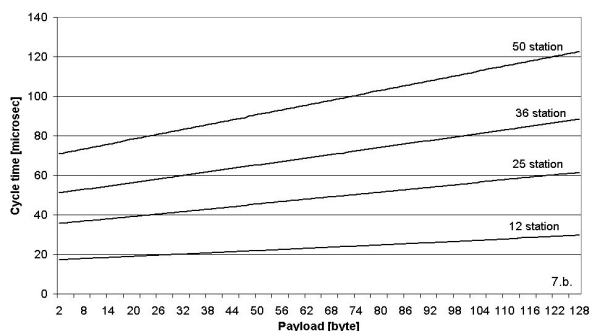
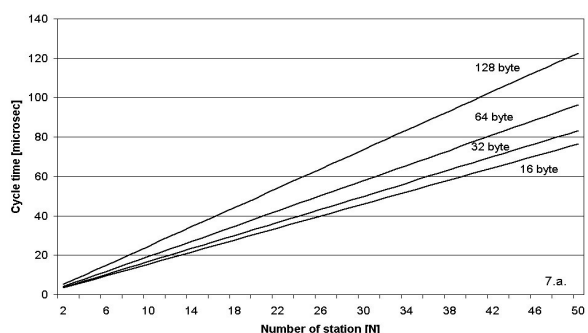
Az előző ábrákból egyértelműen látszik, hogy mindkét meghatározó tényező (N, Data) egyértelműen befolyásolja a ciklusidő növekedését. A növekedés mértéke azonban sokkal alacsonyabb, amikor az EtherCAT terhelést növeljük.

Mindkét ábrán, nyomon követhetjük, hogy azonos terhelés mellett, amikor az eszközök számát megduplázzuk, akkor a ciklusidő is kb. duplájára nő. Például 16 bájttal terhelésnél, ha az eszközök számát növeljük 24-ről 48-ra, akkor a ciklusidő 101 μ s-ról 200 μ s-ra növekszik. De ugyanez mondható el 128 bájttal terhelésnél is. Itt is kb. duplájára növekszik a ciklusidő. Viszont, ha a terhelést duplázzuk (pl. 32-ről 64-re), azonos eszközszám mellett, a ciklusidő jóval kisebb arányban növekszik, átlagban mintegy 50%-al. Ebből azt a következtetést vonhatjuk le, hogy az EtherCAT rendszer busz frissítési ciklusideje kevésbé érzékeny a vezérlési adatok nagyságának a változására. Ez abból is adódik, hogy a ciklusidő egy jelentős összetevője nagymértékben függ az eszközök által meghatározott késleltetési időtől.

Azt is láthatjuk, hogy 128 bájttal terhelésnél, már csak körülbelül 36 eszközt tudunk kiszolgálni, hogy ne haladjuk meg az 500 μ s-os korlátot, amely 50-50%-os időréssel (50% EtherCAT, 50% UDP/IP adatok) összesen, a szigorúan valós idejű rendszereknél a maximális 1ms-os buszfrissítési időt adja.

A továbbiakban azt vizsgáltam, hogy milyen hatással van a ciklusidőre a sebesség növelése. Várhatóan a keretek továbbítási ideje 10-szeressé csökken. Az eszközök késleltetési ideje viszont csak kb. a felére, így $t_D = 0,85\mu$ s értékkel számolhatunk [4].

A következő grafikonok (7. ábra) elemzéséből láthatjuk, hogy jelentősen, mintegy öt és félszeresére csökkent a ciklusidő, viszont a görbék jellege nem változott. A 7.a. ábrán láthatjuk, hogy ugyanúgy duplájára nő a ciklusidő, amikor az eszközök száma megduplázódik. Viszont a 7.b. ábrán azt is megfigyelhetjük, hogy ha az EtherCAT adatokat duplázzuk meg, például 32-ről 64-re, a ciklusidő már csak kb. 15%-al növekszik.



7. ábra. A ciklusidő alakulása Gigabit Etherneten

7. ÖSSZEFOGLALÁS

Összefoglalva az előzőeket kijelenthetjük, hogy az EtherCAT a ciklusidőt illetően 100Mbps-os hálózatban igen jól viselkedik, főleg kisebb adatterhelések (16, 32, 64 bájttal) esetében. A sebesség növelése viszont nem hozza meg a várt eredményt, mert a ciklusidő csak kb. 5,5-szörös csökkenést ér el. Azt viszont láthatjuk, hogy 128 bájttal, nemhogy 50, de akár 100 eszközt is képes kiszolgálni a fentebb már említett időkorláton belül.

8. KÖSZÖNETNYILVÁNÍTÁS

A bemutatott kutató munka a TÁMOP-4.2.1.B-10/2/KONV-2010-0001 jelű projekt részeként az Európai Unió támogatásával, az Európai Szociális Alap társfinanszírozásával valósult meg.

9. IRODALOMJEGYZÉK

- [1] FERENCZI I., Methods to determine a Profinet IRT bus cycle time, MicroCAD Conference 2011.
- [2] M. ROSTAN, EtherCAT Introduction, Hannover messe 2008.
- [3] EtherCAT – The Ethernet Fieldbus, 2009, www.ethercat.org
- [4] JASPERNEITE J., SCHUMACHER M., Limits of increasing the performance of Industrial Ethernet Protocols, EFTA 2007 Proceedings, 17-24 old.