

**Pázmány Péter Katolikus Egyetem
Információs Technológiai Kar
Műszaki Informatika Szak**

Recsnyik Zsolt

***Gerincsérültek végtagmozgatását segítő módszerek
Funkcionális Elektromos Stimuláció segítségével***



2008

**Témavezető: dr. Laczkó József
Műszaki konzulens: Tihanyi Attila**

Alulírott, **Recsnyik Zsolt** a Pázmány Péter Katolikus Egyetem Információs Technológiai Karának hallgatója kijelentem, hogy ezt a diplomatervet meg nem engedett segítség nélkül, saját magam készítettem, és a diplomamunkában csak a megadott forrásokat használtam fel. Minden olyan részt, melyet szó szerint, vagy azonos értelemben, de átfogalmazva más forrásból átvettem, egyértelműen a forrás megadásával megjelöltem. Ezt a Diplomamunkát más szakon még nem nyújtottam be.

.....

Recsnyik Zsolt

Tartalomjegyzék

TARTALOMJEGYZÉK	3
BEVEZETÉS	4
A FELADAT ISMERTETÉSE.....	4
FUNKCIONÁLIS ELEKTROMOS STIMULÁCIÓ (FES).....	4
INGERLÉSI TECHNIKÁK.....	5
NAPJAINK FEJLESZTÉSEI.....	6
FES ÉS TRICIKLIZÉS.....	7
KUTATÁS ÉS REHABILITÁCIÓ	9
MÉRÉS LEÍRÁSA.....	9
ESZKÖZÖK TECHNIKAI ISMERTETÉSE.....	10
<i>Ergométer</i>	10
<i>Stimulátor</i>	10
A MÉRÉSI MÓDSZER.....	11
AZ ÁRAMKÖR	13
MOTIVÁCIÓ.....	13
A TERVEZETT ÁRAMKÖR FELÉPÍTÉSE.....	14
EGYSÉGEK ISMERTETÉSE.....	16
<i>A szenzor</i>	16
<i>Az erősítő</i>	17
<i>A mikrokontroller</i>	18
EGYSÉGEK ILLESZTÉSE, TERVEZÉS, MÉRETEZÉS, TESZTELÉS.....	18
AZ ÁRAMKÖR FIZIKAI MEGVALÓSÍTÁSA.....	26
PIC PROGRAMOZÁS	27
A FEJLESZTÉSHEZ HASZNÁLT KÖRNYEZET ÉS ESZKÖZÖK.....	27
<i>Mplab IDE</i>	27
<i>Projekt Menedzser (Project Manager)</i>	28
<i>Hibakereső (debugger)</i>	28
<i>Az Assembler</i>	29
<i>A mikrokontroller csatlakoztatása</i>	29
A FELDOLGOZÁS ELMÉLETI ALAPJAI.....	30
A MIKROKONTROLLER FELHASZNÁLT EGYSÉGEI.....	32
<i>Konfigurációs regiszterek</i>	33
<i>Timer</i>	33
<i>Interrupt kezelés</i>	35
<i>Soros kommunikáció</i>	37
<i>Adat címzési módok</i>	39
GYAKORLATI MEGVALÓSÍTÁS.....	41
<i>Időmérés a timerrel</i>	41
<i>Adatreprezentáció</i>	42
<i>Gyűrűs regiszter</i>	44
<i>Korlátvizsgálat</i>	46
<i>Státusz regiszterek</i>	49
ILLESZTÉS A STIMULÁTORRAL.....	49
TOVÁBBI FEJLESZTÉSI LEHETŐSÉGEK	51
ÖSSZEFOGLALÁS	53
IRODALOMJEGYZÉK	54

Bevezetés

Két fejezetcím között egy-egy szövegrészt kell írni!

A feladat ismertetése

A diplomafeladat kitűzése szerint ez a diplomaterv a mesterséges izom- és ideg ingerlés technikájával foglalkozik. Munkámban ismertetem az elektromos stimuláció technikáját, működését, eszközeit. Bemutatom a tudomány ezen a területén zajló aktuális, illetve már befejezett kutatásait és az ezekkel elérhető eredményeket. Beszámolok egy aktuálisan zajló kutatás menetéről és az ebből nyert tapasztalataimról, melyek kiindulási alapként szolgáltak egy korszerű mérnöki alkalmazás kifejlesztéséhez. Leírom az alkalmazás fejlesztésének menetét, az így elkészült eszköz célját, működését, technikai paramétereit.

Funkcionális Elektromos Stimuláció (FES)

A funkcionális elektromos stimuláció napjaink egyik intenzíven kutatott alkalmazása. Rendkívül izgalmas interdiszciplináris terület, melyben a biológiai és orvostudományi ismeretek mérnöki eszközökkel és komoly matematikai apparátussal összefonódva alkotnak egységet.

Ez a technika elektromos áram alkalmazásával izom-aktivitás előidézését valósítja meg a központi idegrendszer beavatkozása nélkül. Mindez annak érdekében történik, hogy a funkcionalitásukat veszített, sérült idegpályák által beidegzett izmokat különböző mozgásokra késztessek. Ebből látszik, hogy olyan idegrendszeri sérülések esetén alkalmazzák, melyek bénulást vagy károsodást okoztak a szervezet motoros funkcióiban. Az egyik leggyakoribb ilyen eset a gerincvelő sérülése, mely során idegpályák szakadnak meg, vagy veszítik el funkcionalitásukat, így okozva részleges vagy teljes végtagbénulást. A legtöbb ilyen esetben a szenzoros funkciók is károsodnak, melyek lehetővé teszik az izmok egyébként fájdalmas, áram általi ingerlését.

Klinikai alkalmazásai meglehetősen sokrétűek, számtalan alapot szolgáltatva kutatások számára. Általánosságban azt veszik célba, hogy hogyan lehet visszanyerni izomerőt teljes vagy részleges bénulás esetén, illetve hogyan tudnak

FES alkalmazásával gerincsérült páciensek segítségére az elveszett funkcionalitást pótló eszközöket létrehozni.

A nagyvilágban számos kutató illetve gyártó foglalkozik FES ingerlésre alkalmas eszközök gyártásával és új ingerlési technikák, hatékonyabb módszerek kidolgozásával. Ezek feladatai közé tartozik a pontos fiziológia hatások megismerése, a végtagok modellezése és ezek tökéletesítése, és gyakorlatilag maga a mérnöki megvalósítás – eszközök, algoritmusok, módszerek és implementációik – létrehozása és feljavítása.

Szerepe a rehabilitációban is meghatározó jelentőségű [1]. Elgyengült, régen nem mozgatott vagy nem mozgatható izmok erősítése, edzése tipikus funkcionális elektromos stimulációval megoldható feladat.

Ingerlési technikák

A funkcionális elektromos stimulációhoz szorosan kötődő téma az ingerlés kivitelezése. Az én dolgozatomban is hangsúlyos, mivel a végkifejlethez elengedhetetlen a technikáinak és módszereinek ismerete. A rohamosan fejlődő világnak ezen a területén is számos technika alakult ki elektromos ingerlés végrehajtására. Az elektromos ingerlés gyakorlatilag áram hatására létrejövő funkcionális változást jelent izomsejtekben (vagy idegsejtekben). Ami mindenképpen közös a módszerekben, hogy elektródákkal történik. Az ingerlés helye szerint megkülönböztetünk intracelluláris és extracelluláris ingerlést. Az előbbi a sejten belüli közvetlen stimulációt jelöli, aminek megvalósítása a sejtek méretét figyelembe véve technikailag nehéz feladat. Utóbbi a sejt közvetlen környezetének stimulációját jelenti, mely közvetve hat a sejtekre, könnyebb megvalósítani viszont indirekt befolyásolást takar. Ennek az ingerlő elektródák száma szerint két fajtája létezik. Ezek a monopoláris és bipoláris ingerlés, melyeknél rendre egy illetve két elektróda használatos. Utóbbi koncentráltabb stimulációt tesz lehetővé.

Az elektródák anyaguk szerint különböző áram és feszültség átviteli karakterisztikával jellemezhetők, melyek ez alapján polarizálható és nem polarizálható elektródákra bonthatók. A nem polarizálható elektródák nem változtatják meg az ingerlés jelalakját, ezért jól használhatók – ilyen az ezüst-kloridból készült elektróda.

Az ingerlő elektróda helye szerint megkülönböztetünk beszúrt (beépített), illetve felszíni elektródával történő ingerléseket. A felszíni ingerlés nem invazív módszer, könnyen használható, nem költséges a többi ingerléshez képest, és nem áll fenn az elfertőződés veszélye, mint az invazív technikák esetén, ugyanakkor a kisebb – és legfőképpen a mélyebben fekvő – izomrostok szelektív stimulálása nem megoldott. Ezt az ingerlési technikát általában a petyhüdt izmok erősítésére valamint izomgyakorlatok koordinálására használják. Ezt a technikát alkalmazzuk FES biciklizés során.

Ami a későbbiekben is fontos szerephez jut, hogy az ingerlés nem folytonos áraminjektálással történik, mivel ennek súlyos szövetroncsoló hatásai lehetnének hosszabb alkalmazás után. Tehát az ingerlés a motoneuronok tüzeléséhez hasonlóan repetitív impulzusokkal történik, ami az ingerlési időtartam alatt meghatározott frekvenciával történő stimuláló impulzusok generálását jelenti. Itt a polaritás szempontjából megkülönböztetünk mono – és bifázisos ingerlést. Utóbbinak előnye, hogy a töltésinjekció konstans mértékű az ingerlés alatt, a töltések ide-oda haladnak az elektródák között (ezáltal elkerülhető a szövetek töltésfelhalmozódás okozta elektrolízisének a veszélye).

Napjaink fejlesztései

Napjaink fejlesztési irányait figyelembe véve számos innovációt találhatunk. Többek között léteznek markolást segítő rendszerek, melyek féloldali bénulás (hemiplegia) vagy minden végtag bénulása (tetraplegia) esetén a kéznek ezt a kiesett funkcióját teszik lehetővé FES segítségével. Egy clevelandi fejlesztésű Freehand System nevű készülék a fentebb említett képességcsökkenéssel élők mindennapi életét könnyíti meg. Erős markoló funkciót valósít meg - relatíve nehéz dolgok felemelését segíti -, így téve könnyebbé és függetlenebbé felhasználói életét [2].

Találhatunk részleges lábbénulás hatásaként jelentkező úgynevezett dropped foot probléma megoldására konstruált szerkezetet, mely szintén FES segítségével működik. A dropped foot probléma izomgyengeség vagy bénultság okozta jelenség. A járás során a lábfejet emelő izmok nem működnek (vagy gyengék), állandó botladozással vagy akár eleséssel fenyegetve a haladást. A kifejlesztett rendszer

felszíni elektródák használatával stimulálja a lábfej emelésében szerepet játszó izmokat a helyes működés érdekében.[3]

Nagyon érdekes és futurisztikusnak tűnő szerkezetekkel is találkozhatunk, mint pl. a BION-nak nevezett beépíthető szerkezet. Számos alkalmazása lehetséges, mivel fejlett technikájának köszönhetően érzékelőként és beavatkozóként is képes működni. Több fajtája létezik, ezek közül van, amely mindössze 2 mm vastag. Számos klinikai rendellenesség kezelésére használják sikeresen, mint például a fentebb említett esetek, vagy váll megereszkedés és izom atrofia [4].

Nagy számban fellelhetők a FES rehabilitációban történő alkalmazásai is. Találhatunk a járás újratanulását segítő készüléket is. De ezek közül kiemelt fontosságú számomra a funkcionális stimulációval segített kerékpározó mozgás, amely a dolgozatom témájának alapja.

FES és triciklizés

A FES - nem kizárólag a biciklizés, hanem egyéb más FES segítségével támogatott tevékenység - hasznossága és hatékonysága az izmok erejének és kitartóságának növekedésében, izommerevség csökkenésében és a végtagok keringésének javulásában is megmutatkozik. Nem utolsó sorban hatása a páciensek pszichológiai állapotára nézve is jótékonynak bizonyul.

A paraplégiás emberek általában tolószék segítségével mozognak, mivel felső testük izmai teljesen egészségesek. Ennek egyetlen hátránya, hogy a teljes mozgási energiát a felsőtestük izommunkájával érik el, így a láb izmai, melyek kapcsolata a központi idegrendszerrel megszakadt egy gerincvelői sérülés során, az idő elteltével egyre gyengébbek lesznek.

Összehasonlítva más mozgásokat előidéző gyakorlatokkal, a biciklizés azzal az óriási előnnyel rendelkezik, hogy a pedálra kifejtett erő nagy hatékonysággal alakul át mozgási energiává, és a meghajtása lábbal történik. Ezért a nem használt lábizmok egyik legjobb edzési módszere a FES biciklizés. [5]

A FES triciklizés több módon is megvalósítható, de vannak általános elemek, amik elengedhetetlen feltételei. Más-más célokra is találhatunk FES triciklizési konstrukciókat. Egyesek normál triciklit használnak edzési és rehabilitációs célokra, mások ergométereket. Ráadásul az ingerlési mintázatok meghatározása sem egy

standard, meglévő, kialakult módszer alapján történik. Úgy is lehetne mondani, mindenki másképp csinálja.

Ami általánosan elfogadott, hogy a tekerésre alkalmas eszköz valamilyen formában meghatározza az ingerlési mintázatot. Általában a pedál állása az információ, ami ehhez szükséges [6]. Ezen felül azonban komoly mérnöki feladat a részletek meghatározása, egy jól működő rendszer megalkotásához. A diplomamunkámban ezt a feladatot oldom meg. A kiindulópontot a következőkben leírt kutatás adja.

Kutatás és rehabilitáció

Két fejezetcím között egy-egy szövegrészt kell írni!

Mérés leírása

Néhány évvel ezelőtt az Országos Orvosi Rehabilitációs Intézetben (OORI) elkezdődött egy funkcionális elektromos stimulációval segített rehabilitációs program, melyben a PPKE hallgatói is részt vettek és vehetnek a mai napig. Általában az OORI-be kerülnek az ország legnagyobb részéről a gerincsérülést szenvedett páciensek. Aztán itt tanulják meg, hogyan kezeljék a sérülésükből adódó nehézségeket, és egészségi állapotukra jó hatással lévő különböző edzési programokon vesznek részt. A rehabilitációs program keretében FES-sel segített biciklizést is végeznek.

Személyes tapasztalataim alapján nagyon szívesen jönnek el az edzésekre, hiszen ők is érzik a FES hatékonyságát, mint ezt tudományosan is igazolták szintúgy az egyetem hallgatói, mint a nagyvilág kutatói.[7]

A megközelítőleg – az előkészületekkel együtt – egy óra időtartamú edzéseken egyidejűleg egy páciens vesz részt, melynek során megtörténik a kísérleti összeállítás felszerelése, egy bemelegítési fázis, a stimulációval segített tekerés, és egy levezetési fázis. A kísérleti összeállítás legfontosabb eszközei az ergométer, amely a tekerés nehézségi fokozatának szabályozására és a tekerés minőségére vonatkozó néhány paraméter megfigyelésére alkalmas és a stimulátor, amelynek segítségével az izmok ingerlése történik. Az edzéseken az ergométer és más orvosi eszközök által regisztrált, a kutatás részét képező adatok közé tartozik az átlagos teljesítmény és csúcsteljesítmény, az egyes nehézségi fokozatokban történő tekerés időtartama és átlagos fordulatszáma, a teljes megtett út hossza, a páciens tekerési mintázatára vonatkozó szimmetria, illetve a páciensek állapotát monitorozandó a vérnyomás és a pulzusszám is [8] [9] [10]

Eszközök technikai ismertetése

Két fejezetcím között egy-egy szövegrészt kell írni!

Ergométer

A MotoMed Viva2 típusú ergométer (**1.ábra**) egy multifunkcionális készülék, melynek segítségével biztonságos módon edzhetnek bénult, izommerevséggel és izomgyengeséggel küzdő betegek. Egyszerűen kezelhető menüvel, nagy LCD kijelzővel rendelkezik. Beépített motor segítheti a használó munkáját bemelegítés és edzés során. Különböző nem várt, görcsös izom-összehúzódásokat is tud biztonságosan kezelni. Az edzési adatok rögzítésére és ezek analizálására is képes. Mindemellett meghatározható vele a pedálozás szimmetriája is, tehát hogy mennyire oszlott meg a tekerés közben fellépő terhelés a két végtag között.

Stimulátor

A MotionStim8 nevezetű stimulátor (**1.ábra**) egy beépített akkumulátor segítségével működő készülék. 8 csatornán képes különböző ingerlő impulzusok kiadására. Minden egyes csatorna egymástól függetlenül, maximálisan 125 mA nagyságú, áramimpulzus generálására képes, miközben az impulzusszélesség 10 és 500 mikrosec, a frekvencia 1 és 99 Hz között állítható. Ezen tulajdonságok mindegyike 1 lépésköz pontossággal hangolható. A generált impulzusok négyszögjel alakú bifázisos jelek. A stimulátor rendelkezik soros portos bemenettel. A gyártó által biztosított szoftver segítségével egyszerű ingerlési mintázatok hozhatók létre és tárolhatók memóriájában. Szintén folyadékkristályos kijelzővel rendelkezik. Kezelése nyomógombok segítségével történik, minden csatornára külön áramerősség-szabályozóval.



1. ábra: MotoMed Viva2 ergométer és MotionStim8 stimulátor

A mérési módszer

Az eddigi ingerlési módszert az egyetem hallgatói nemzetközi együttműködésben állították össze a kutatások eredményeiből [11] [12]. A lényeg a következő: a biciklizéshez szükséges legnagyobb izmokat, a combfeszítő és combhajlító izmokat ingerli a stimulátor, melynek 8 csatornája közül erre a célra csak négy van felhasználva. Nagyon fontos, hogy a stimulátor és az ergométer egymással kompatibilis eszközök ezért tudnak összehangoltan működni. Az ergométer által mért adatok alapján történik az izmok ingerlése a megfelelő időpillanatban. Ehhez feltételezhetően a legfontosabb adat az ergométer pedáljának állása (abszolút szögpozíciója). Ennek ismeretében lehet beállítani a stimulációs mintázatot a stimulátor felprogramozásával. A stimulációs mintázat esetünkben egészséges emberek részvételével végzett kísérletek alapján lett kialakítva, melynek hatékonysága a teljesítmény adatokra nézve sokkal jobbnak bizonyult, mint a gyári program.

Az eddigi mérésekben rendelkezésre álló eszközök az ergométer és a stimulátor. Ezek együttműködésének eredményeképp vált lehetségessé a FES alkalmazása rehabilitációs célokra az OORI-ben. Itt jellemzően a kísérletek helyhez kötötten zajlanak. A munkám célja a helyhez kötött ergométer lecserélése bármely alkalmas triciklire, hogy így lehetségessé váljon a helyváltoztatásra is alkalmas edzés. Ennek eredményeképp új paraméterek vizsgálatára is lehetőség adódhat, mivel az ergométer belső, a tekerésre jellemző adatait nem lehetett eddig kinyerni. Másik óriási előny az lehet minden egyes gerincsérülést szenvedett beteg számára, hogy

nem kell horribilis összegeket kifizetni egy ergométerért, hanem egy újonnan kifejlesztett eszköz használatával bármely tekerésre alkalmas jármű segítségével edzhetnek.

Mindezek elérésének esszenciális lépése egy olyan eszköz létrehozása, melynek segítségével kiváltható az ergométer funkcionalitása. Ehhez elsősorban az ergométer által szolgáltatott, az elektromos ingerléshez szükséges adatokra van szükség, mely alapján a stimulátor működhet. Az egyik legfőbb ilyen adat az ergométer pedáljának szögállása. Ez alapján lehet felprogramozni a stimulátort, hogy a különböző pedálálláshoz (vagyis gyakorlatilag az alsó végtag különböző elhelyezkedéséhez) tartozó ingerlő impulzusokat a megfelelő ingerlési csatornára adja.

Az áramkör

Két fejezetcím között egy-egy szövegrészt kell írni!

Motiváció

A kiindulási pont tehát az OORI-ben felállított kísérleti mérési elrendezés, melyet a következő sematikus ábra szemléltet (**2.ábra**).



2. ábra: Stimulációs elrendezés a rehabilitáció elősegítésére

A stimulátorhoz 2004 – ben készítettek egy leírást, amelyben közlik, hogy a készülék külső jelek alapján történő vezérlése lehetséges soros porton keresztül, úgynevezett ScienceMode – ban. Ennek két lehetséges formája van: a Channel List Mode, amely komplex ingerlési minták küldését teszi lehetővé, illetve a Single Pulse Mode, amely utasítások küldésével vezérli a stimulátort, hogy melyik csatornáján, milyen erősségű és szélességű impulzust adjon ki. A dokumentáció részletesen ismerteti a kommunikáció beállításának paramétereit, illetve a kiadható utasítások szintaktikáját és értelmezését.

Ennek értelmében az ergométer abban az esetben cserélhető le, ha valamilyen formában rendelkezésre áll számunkra az adott jármű pedálállásának szöginformációja, illetve ez alapján képesek vagyunk a megfelelő ingerlési mintázat előállításához szükséges parancsokat továbbítani a stimulátor felé.

Egy tricikli pedálszögének megállapítására számos módszer lenne alkalmazható. Ebben az esetben mindenképpen szem előtt kell tartani a végső célt, miszerint egy elektromos berendezést (stimulátort) szeretnék ezzel az információval vezérelni. Tehát célravezető lehet elektromos formában reprezentálni a pedál szögét vagy valamely erre utaló tulajdonságot, amelyből egyértelműen következik a pedálszög.

Ezért szükség lesz valamilyen szenzorra, amely egy fizikai állapotot érzékelve elektromos jelet állít elő. Ennek az elektromos jelnek a kondicionálása, értelmezése és feldolgozása a feladat egyik legtöbb figyelmet érdemlő része. Ennek során alakul ki a szenzor jeléből a pedálszög reprezentációja. A megfontolásokat figyelembe véve az elérendő cél a következő sematikus ábrán látható (**3.ábra**).



3. ábra: Tervezett stimulációs elrendezés

Fontos részfeladat a stimulátorral való kommunikáció megoldása – a már valamilyen formában rendelkezésre álló pedálszög információ alapján.

Az elsődleges cél a pedálszög információ előállítása, ezt a feladatot takarja a diplomamunka, a megvalósítás további lépéseire csak javaslatot és elméleti megoldást szolgáltatók.

A tervezett áramkör felépítése

A kitűzött célt megvalósító áramkör magába foglalja a szenzoros jel kondicionálását, feldolgozását és a stimulátorral történő kommunikációra is szolgáltató megfelelő eszközöket. Az előfeltételeket figyelembe véve az érzékelés egyik legjobb formája, ha a hajtókarhoz közvetlenül rögzített fém fogaskerék egyes fogait tudjuk érzékelni. Ez egyértelmű leképezése a hajtókar szögpozíciójának, ha találunk megfelelő referencia pontot. Így diszkrét pontossággal meg lehet határozni a pedálszög állását. A pedál egy teljes körülfordulása alatt pontosan annyi fog halad el a szenzor érzékelő terében, amennyi van a fogaskeréken. Ezt megszámlálva és kijelölve egy referenciafogot, pontosan 360/darab fokos pedálszög változást tudunk érzékelni. Tehát ilyen pontosan tudjuk megmondani, hogy a referencia fogtól (ami impliciten

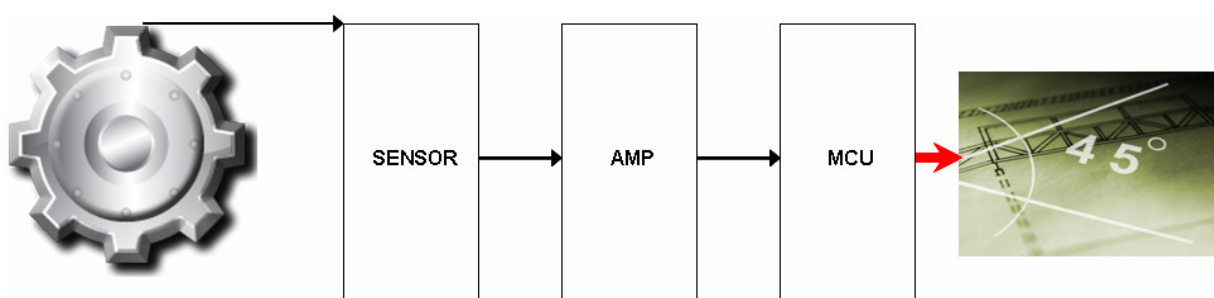
meghatároz egy pedálpozíciót) hány fokra áll a pedál. Ez a referencia fog egy jelölt fog, amelyet az érzékelő jele alapján 100%-os biztonsággal meg tudunk különböztetni a többitől. Az én megoldásomban ez egy lereszelt fog, amely esetben nem történik szenzoros érzékelés. Ez a megoldás kézenfekvő, hiszen egy fog lereszelésével a fogaskerék továbbra is megbízhatóan tud működni. Ennek a módszernek az alkalmazásával viszont figyelni kell az egyes szenzoros érzékelésből származó jelek között eltelő időt, hogy meg tudjuk állapítani melyik pillanatban kellett volna a hiányzó jelnek érkeznie az érzékelőn.

Ezek alapján az elkészítendő áramkörnek képesnek kell lennie a következőkre:

- A bemeneti szenzoros jel kondicionálása (leginkább erősítésről van szó)
- Megállapítani a bemenő jelből, hogy az egy különálló impulzus
- Megszámolni ezeket az impulzusokat
- Mérti az impulzusok között eltelő időt
- Eltárolni szükséges adatokat
- Műveleteket végezni ezekkel az adatokkal
- Megfelelő módon reprezentálni az adatokat
- Megfelelő protokoll alapján történő kommunikációt megvalósítani

A fenti követelményeket egy mikrokontroller segítségével messzemenően ki lehet elégíteni. A jel előerősítését egy kis áramkör valósítja meg, amelyből az erősített jeleket a mikrokontroller értelmezi, dolgozza fel és alakítja át megfelelő formára.

Az elkészített áramkör blokkvázlata az alábbi ábrán látható (**4.ábra**)



4. ábra: A fejlesztett áramkör blokkvázlata a fogaskeréktől az abszolút szögpozícióig

Egységek ismertetése

Két fejezetcím között egy-egy szövegrészt kell írni!

A szenzor

Érzékelő szenzorokból meglehetősen sok van és számos, különböző megoldás létezik. Ezek közül néhányat fontolóra véve jutottam közelebb a megoldáshoz.

Elsőként infravörös távolságmérő szenzor alkalmazására gondoltam, mellyel egy referenciaponttól mért távolság alapján lehetne a pedál szögét visszaszámolni. Kis matematikai elemzés után kiderült azonban, hogy a megoldás nem lenne egyértelmű egy referencia pont megadásával. Felmerültek olyan technikai problémák is, mint referenciapontok száma és megadásának módja, pontosság.

Számításba jött lézeres távolságmérés de ez az előző megfontolások miatt (a pontosságot leszámítva) szintén nehezen lett volna alkalmazható. Így jutottam arra a következtetésre, hogy a mágneses alapú érzékelés lehetősége jóval közelebb áll egy megfelelő megoldáshoz. Mivel a fogaskerekek legtöbbször mágnesezhető fémből készül, ezt kihasználva jól érzékelhetők a fogaskerék fogak is

Ezért végül egy mágneses elvű speciális detektor mellett döntöttem. Ezt a szenzort autóknál alkalmazzák és a neve felső holtponti jeladó, melyet robosztussága – a környezeti viszonyoktól való függetlensége – és olcsósága miatt választottam. Benzines és dízel autók főtengely pozíciójának meghatározására használják. Az alapelv itt is ugyanaz: a főtengelyen levő fogaskerék egy foga hiányzik, amelynek detektálása meghatározza a főtengely pozíciót. Erre azért van szükség, hogy a gyújtási időt ez alapján és egyéb aktuális rendszerparaméterek alapján úgy tudják beállítani, hogy ezzel maximális motorikus teljesítményt érhessenek el. Előnyei közé tartozik, hogy nem kopik el és a szennyeződések sem akadályozzák a helyes működésben.

A feladathoz sok szempontból megfelelő, mivel egyszerűen kezelhető, nem igényel elektromos táplálást és mérete is megfelelő. Működési elve a mágneses tér megváltozásán alapul, hasonló egy Hall szenzorhoz. Lényege, hogy ha valamilyen mágnesezhető anyag elhalad a szenzor előtt, és elég közel van hozzá akkor elektromos jelet generál.

Ennek a jelnek jellemző a formája: ha a mágnesezhető fém belép a szenzor érzékelő terébe, a lefedettséggel arányos nagyságú feszültség keletkezik, amely felfogható,

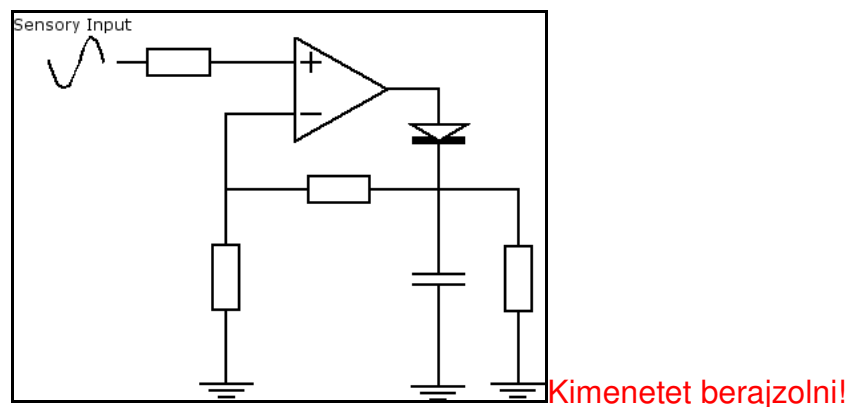
mint egy az előtte elhaladó fémtárgy által generált impulzus. A szenzor a mágneses tér megváltozását érzékeli, tehát ha kilép a fém az érzékelő teréből az előző impulzussal ellentétes polaritású feszültség generálódik. Így a megjelenő jel szinuszos jellegű annak függvényében, hogy a fémtárgy mennyire fedti le az érzékelő szenzor terét.

Ezt az érzékelőt a pedál által hajtott fogaskerékhez elég közel helyezve, képes minden egyes fogat egy impulzussal jelezni. A jeladó áramkörü szempontból passzív elem, nincs szüksége tápellátásra. Emiatt a kibocsátott impulzus kis amplitúdójú, 10 mV-os nagyságrendbe esik. Ennek ellenére zajérzékenysége kicsi, **mivel kis impedanciás az érzékelő.**

Mindezeket figyelembe véve megfelelő mérőeszköz a pedál állásával összefüggésben álló tulajdonság elektromos úton történő kinyerésére.

Az erősítő

A szenzor által kibocsátott jelet nehéz önmagában bármilyen feldolgozásnak alávetni, hiszen nagyon kis feszültségről van szó. Ezért szükséges egy erősítő áramkör, amely lehetővé teszi a feldolgozást. Egy egyszerű kapcsolás elegendő, mely az alábbi módon épül fel (**5.ábra**).



5. ábra: Az előerősítő áramkör kapcsolási rajza

Tartalmaz egy műveleti erősítőt, amely a negatív visszacsatolással az ellenállások arányában erősíti a bemenetén lévő jelet [13]. Az így kapott kimenetet a kondenzátor

segítségével tároljuk, amíg nem érkezik egy újabb impulzus, így gyakorlatilag megkapjuk a pozitív impulzusok burkolóját felerősítve.

A mikrokontroller

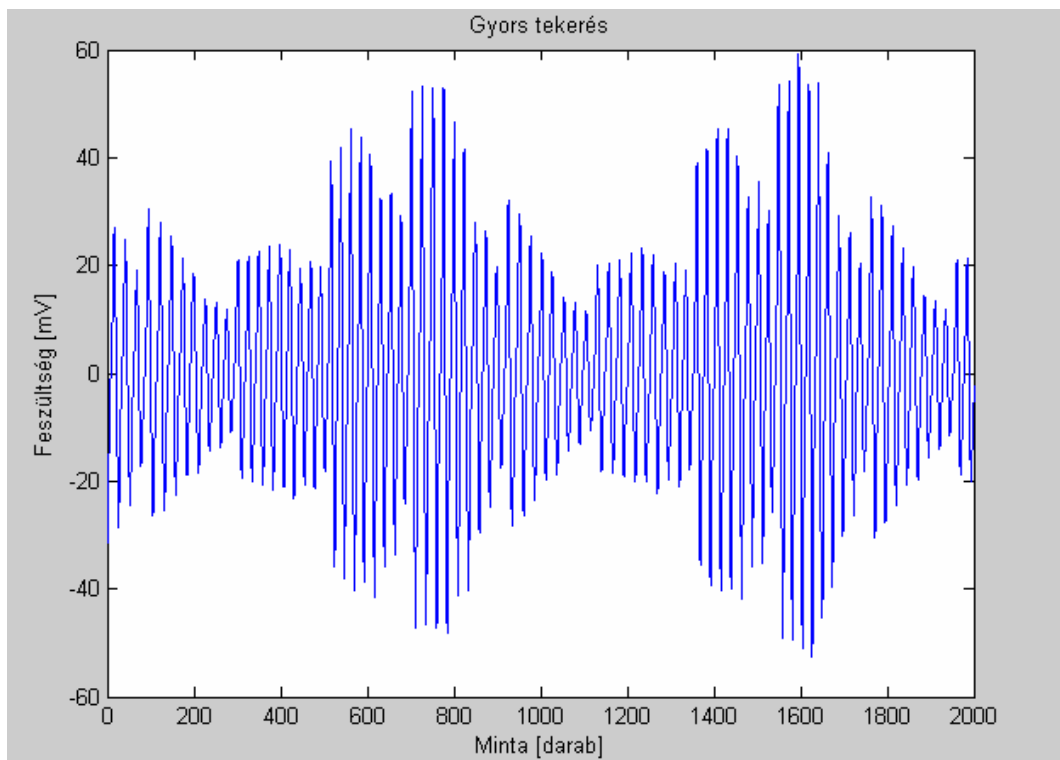
A feladatban meghatározott feldolgozást megvalósító mikrokontroller a MicroChip által gyártott PIC18F2550 típusú 8 bites integrált áramkör. Ennek tulajdonságai a teljesség igénye nélkül a következők:

- Alacsony energiaigényű (Low Power) – célszerűen néhány voltról üzemel
- Timer az időméréshez
- Komparátor belső referenciával
- USART a soros portos kommunikációhoz
- Harvard architektúra - elkülönül a program memória és az adatmemória
- Sok memória - az eltárolandó adatokhoz képest többszörösen túl van dimenzionálva
- Gyors - a feladat által megszabott maximális feldolgozási sebességet többszörösen túlteljesíti
- Programozható - elektronikusan törölhető memóriájába akárhányszor új programok tölthetők
- Adott fejlesztői környezet, amely ingyenesen letölthető és korlátlanul felhasználható

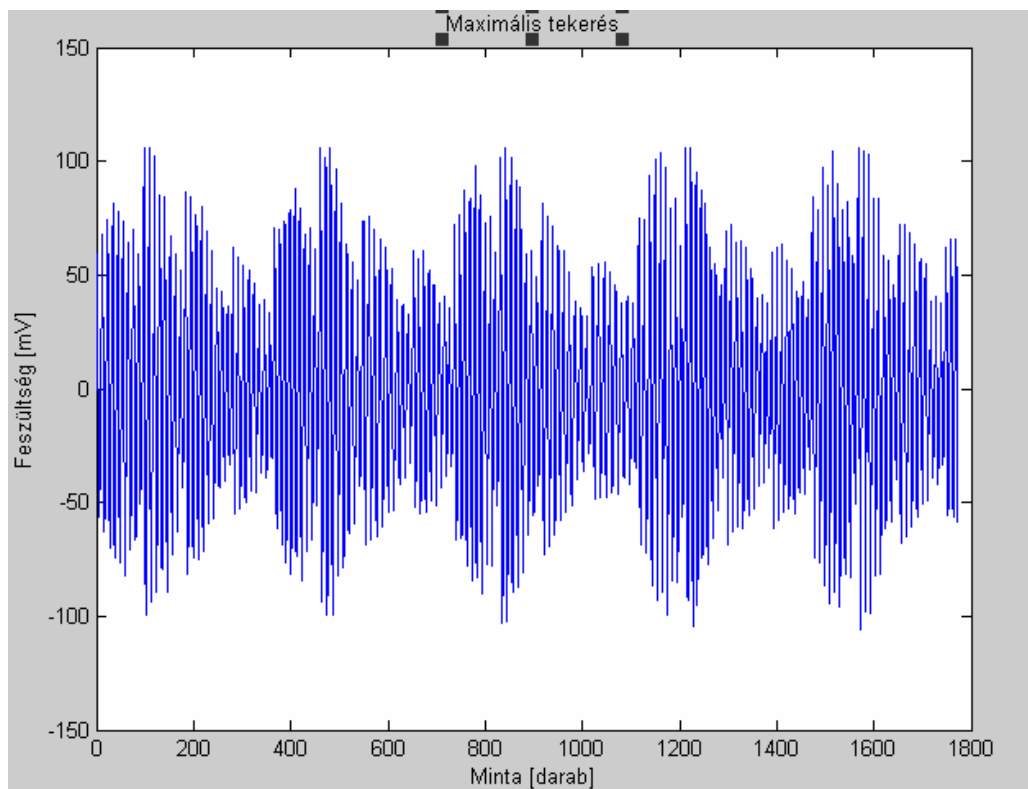
Átrendeztem fontossági sorrendbe

Egységek illesztése, tervezés, méretezés, tesztelés

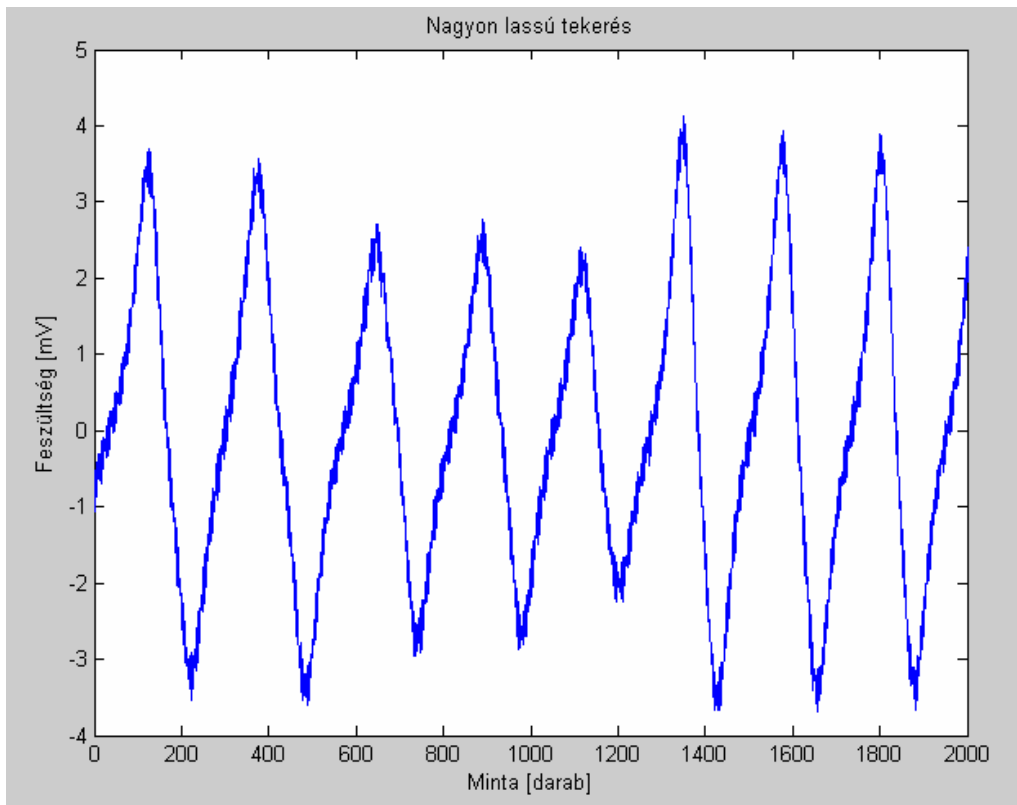
Első lépésként a szenzor tesztelése volt fontos, hiszen ehhez kell igazítani az egész áramkör méretezését. A tekeréstől függő tényezők szintén nagyban befolyásolják a tervezést, ezért a szenzor által kiadott jel paramétereit a triciklire felszerelve mértem. Ehhez a National Instruments Elvis nevű készülékét használtam és a hozzá készült szoftvert. Ez a készülék egy PC-hez csatlakozik és annak az erőforrásai is a rendelkezésre állnak. A programba épített oszcilloszkóp segítségével különböző stílusú elvezetések készítem és tároltam el. Ezek között szerepelt lassú (**8.ábra**), közepesen gyors (**6.ábra**) és nagyon gyors (**7.ábra**) tekerés. Az ábrákon észrevehető az elvezetett jelek néhány jellegzetes tulajdonsága.



6. ábra: Gyors tekerés közben rögzített szenzoros jel amplitúdója a mintaszám függvényében

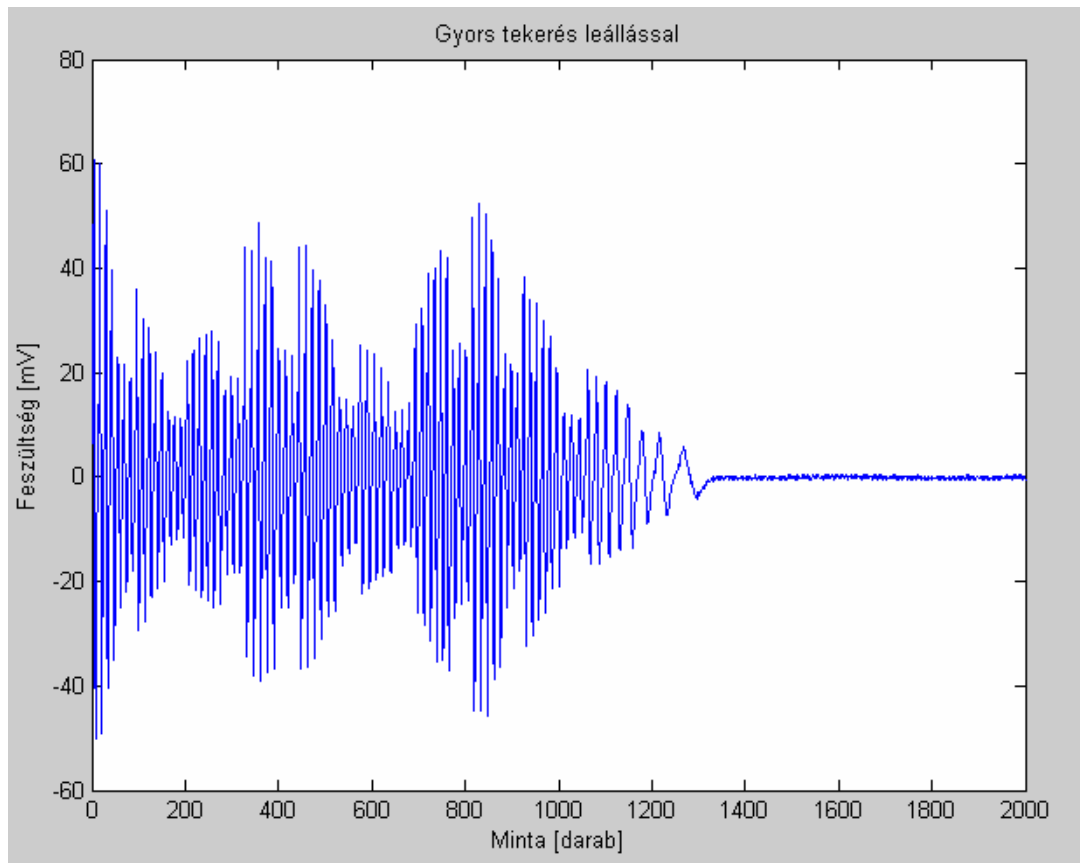


7. ábra: A lehető leggyorsabb tekerés közben rögzített szenzoros jel amplitúdója a mintaszám függvényében



8. ábra: A lehető leglassabb tekerés közben rögzített szenzoros jel amplitúdója a mintaszám függvényében

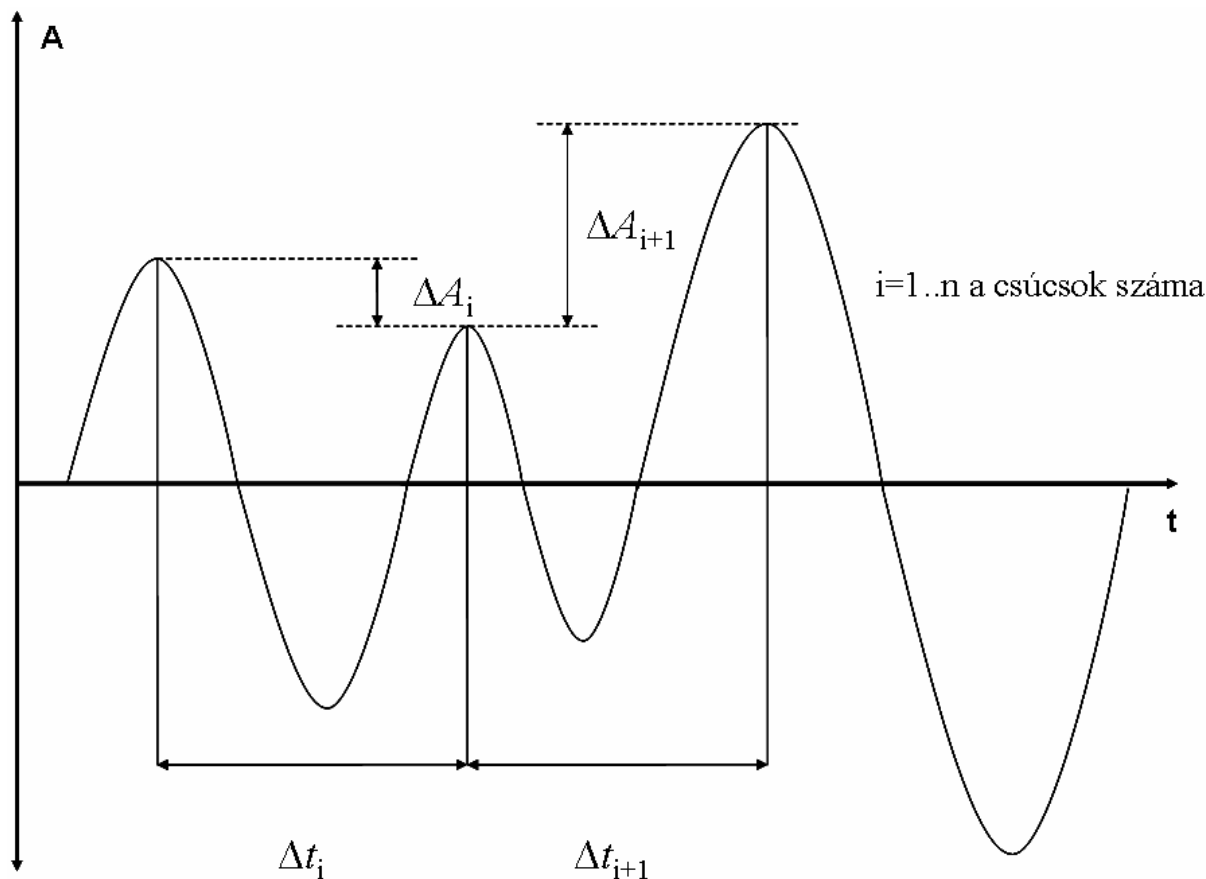
A gyors tekeréseknél látható, hogy amplitúdó moduláltn történik az elvezetés, a fogaskerék tengelyének bizonytalansága miatt. Azaz a fogaskerék úgy mond kileng tekerés közben, és hol közelebb hol távolabb kerül a szenzor érzékelő teréhez. Megfigyelhető, hogy minél gyorsabb a tekerés annál nagyobb amplitúdójú görbéket sikerült rögzíteni. Ez a szenzor működését jellemzi, ugyanis a mágneses tér változása indukálja az impulzusokat, minél gyorsabban történik a változás annál nagyobb a maximális amplitúdó. Az elvezetések amplitúdó tartománya megközelítőleg 10mV és 100mV között van. A leglassabb tekerésnél megfigyelhető, hogy a jel alacsony feszültségtartományban sem lesz bizonytalan. Végül egy utolsó kép, amely egy tekerés dinamikájának változását mutatja be (**9.ábra**).



9. ábra: A tekerés dinamikájának változása közben rögzített jel amplitúdója a mintaszám függvényében

A tárolás sajnálatos módon csak egy viszonylag rövid időablakot jelent, amely alatt a pedál éppen mozog. Ebből aztán a jel néhány releváns jellemzőjét egy saját készítésű MatLab program segítségével határoztam meg.

Fontos jelparaméter a szenzor által tekerés közben kibocsátott maximális amplitúdó. A tekerésre jellemző tulajdonságok pedig a maximális amplitúdóváltozás két egymást követő impulzus esetén, illetve ettől függetlenül a minimális eltelt idő két impulzuscsúcs között (**10.ábra**). Ezek a paraméterek határozták meg ebben az esetben milyen méretezésű áramköri elemekre lesz szükség a jel kondicionálásához.



$$\text{AMP} = \frac{V_{\text{táp}}}{\max_i(A)}$$

$$i(t) = C \frac{dv(t)}{dt} = C \frac{\max_i(\Delta A_i)}{\max_i(\Delta t_i)}, \text{ ahol } v(t) = V_0 \left(1 - e^{-\frac{t}{RC}} \right)$$

és $v(t)$ a kondenzátoron lévő feszültség változását leíró időfüggvény,

AMP az erősítés és $V_{\text{táp}}$ az áramkör tápfeszültsége

10. ábra: Az előerősítő áramkör méretezésére felhasznált fontosabb jelparaméterek és összefüggések

Képleteket pontosan elmagyarázni! A változókat és paramétereket bemutatni!

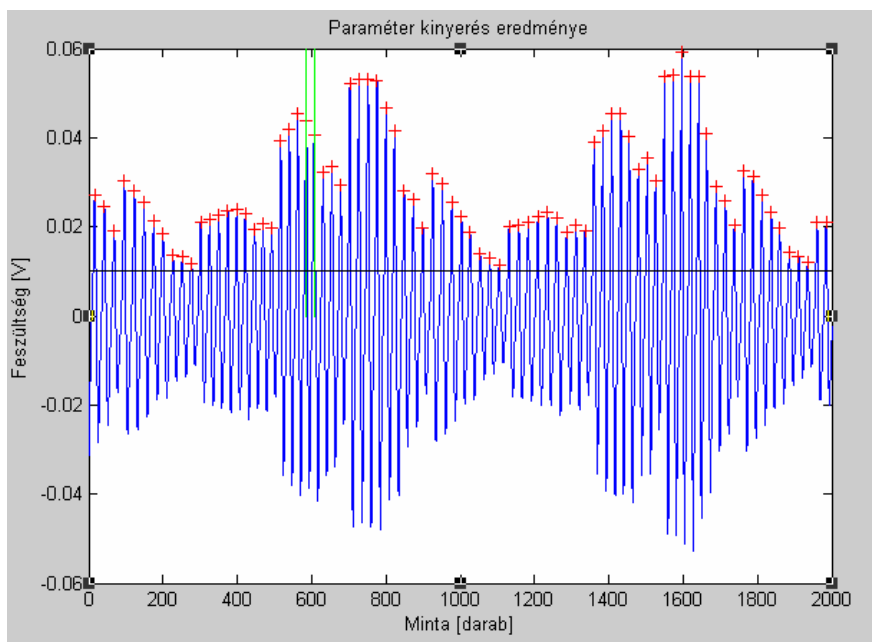
A legfontosabb méretezendő elem a kapcsolatban a feszültséget tároló kondenzátor, amelynek kimenetén gyakorlatilag a felerősített impulzussorozat fog szerepelni. Ennek méretezéséhez kellett meghatározni a tekeréstől függő paramétereket és a maximális amplitúdót.

Az előfeldolgozás lényege a következő: a mV-os nagyságrendű jelből szeretnék egy olyat kapni, amely úgy van kierősítve, hogy a pozitív és negatív tápfeszültség közötti teljes dinamikatartományt kihasználja. Elméletben az előerősítő áramkör egy burkoló

görbét simít rá az eredeti szenzoros jelre, és ezt sokszorosára erősítve adja vissza a kimeneten.

Először ehhez kell ismerni a maximális amplitúdót, amelyet megfeleltethetünk a tápfeszültségnek. Tehát ez érheti el a dinamika tartomány maximumát. Ez az érték az általam rögzített elvezetésekben megközelítőleg 100mv-ra adódott, ami egy 5v-os tápfeszültség esetén (ami a leggyakrabban használatos) minimum 50-szeres erősítés szükségességét vetíti előre.

Másodszor meg kell határozni a kondenzátor kapacitását, úgy hogy mindegyik impulzus megjelenjen a kimenetén. Tehát meg kell határozni milyen gyorsan kell kisülnie és mennyire, hogy egy impulzus csúcsot se veszítsünk el. Ehhez vizsgáltam a legrosszabb esetet, a minimális időbeli eltérést és maximális amplitúdóbeli eltérést két csúcs között. Ennek érdekében írtam egy Matlab programot, amely a korábban elvezetett jelekből meghatározza a szükséges paramétereket. A következő ábrán ennek kimenete látható (**11.ábra**)



11. ábra: A rögzített jel Matlab programmal történt feldolgozás után

A szomszédos csúcsok között minimálisan eltelt idő megközelítőleg 10 ms-ra adódott, a maximális amplitúdó változás közöttük megközelítőleg 25 mV-ra. Ezek az adatok meghatározzák, hogy a kondenzátornak jóval kevesebb mint 10 ms idő alatt tudnia kel töltni 2,5 V-ot, és legalább ennyi idő alatt esnie is ennyit. Mivel a műveleti erősítő által szolgáltatott 20 μA / ps árammennyiség bőven elegendő az ennyi idő alatt történő feltöltődéshez, az 10. ábrán látható képletek alapján egy 1 μF -os kondenzátor tökéletesen kielégíti a jel által szabott feltételeket.

Harmadszor a már korábban feltételezett 50 –szeres erősítést csak a visszacsatoló ellenállások aránya határozza meg, ezért tetszőleges értékűnek választhatók. A dióda szerepe annyi, hogy ne folyjék a műveleti erősítő irányába – azaz visszafelé – áram.

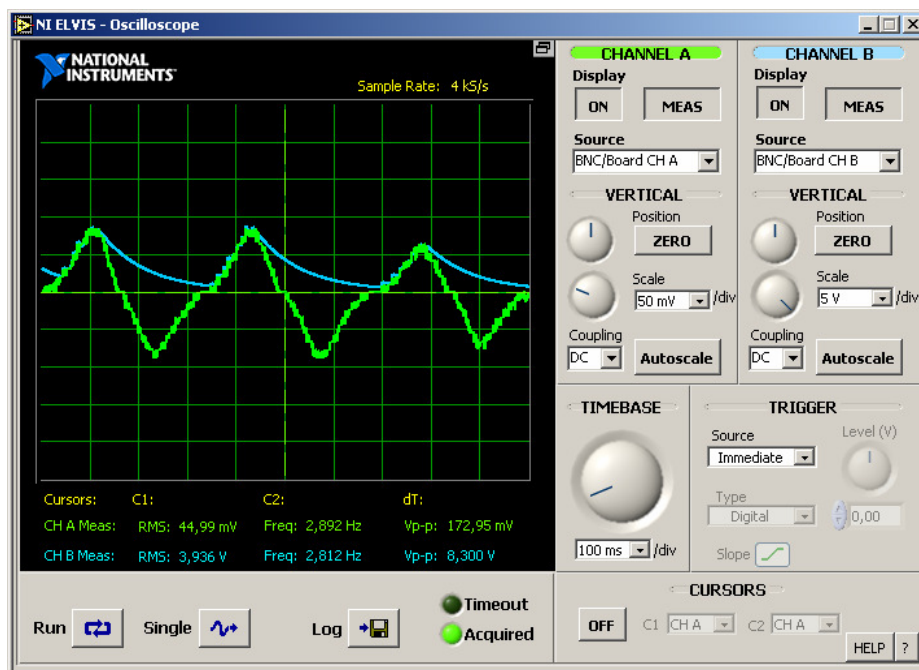
A műveleti erősítő funkcionalitás szempontjából nem befolyásolja a működést, ezért bármikor bármely IC-re cserélhető. Egyetlen fontos paramétere, hogy 5V-os tápfeszültséggel működjön, mert a magasabb tápfeszültség a szükséges elemek száma miatt a hordozhatóság rovására menne.

Az előre számolt minimálisan 50-szeres erősítés sajnos azonban nem elegendő, mert a nagyon alacsony feszültségeket kevésbé erősíti ki, ez a későbbiek során okozott problémát, ezért jóval nagyobb visszacsatolás vált szükségessé.

Az előfeldolgozó áramkör méretekkel együtt már megvalósítható. A teszteléshez szintén felhasználtam az Elvist, amely sokban leegyszerűsíti ezt a lépést. Egyszerűen és logikusan felépített panelje segítségével könnyen előállítható a tesztelni kívánt áramkör. Az összeépítés után a tesztelést a korábban rögzített szenzoros jellel végeztem a tricikli mobilizálásának körülményessége miatt. Ehhez felhasználtam a beépített függvénygenerátort, amely az oszcilloszkóp által rögzített jelet néhány nehézség után képes valamilyen formában generálni. A problémát az jelentette, hogy a szöveges fájlba kiírt, oszcilloszkóp által generált jelet a beépített függvénygenerátor más elrendezéssel várta. Ezt egy újabb Matlab függvény segítségével küszöböltem ki. Itt bukkant fel az a szemléleti eltérésből eredő hiba, hogy a MatLab linuxos sorvége jelet használ, míg az Elvis az operációs rendszer által meghatározottat, ami kis jelentőségű de bosszantó tény. A Matlab program is a melléklet része.

Tehát az eredetileg analóg szenzoros kimenetet, egy analóg/digitális konverzió után eltárolva, majd előállítva ez alapján egy függvénygenerátor segítségével sikeresen elvégezhetővé vált az áramkör vizsgálata.

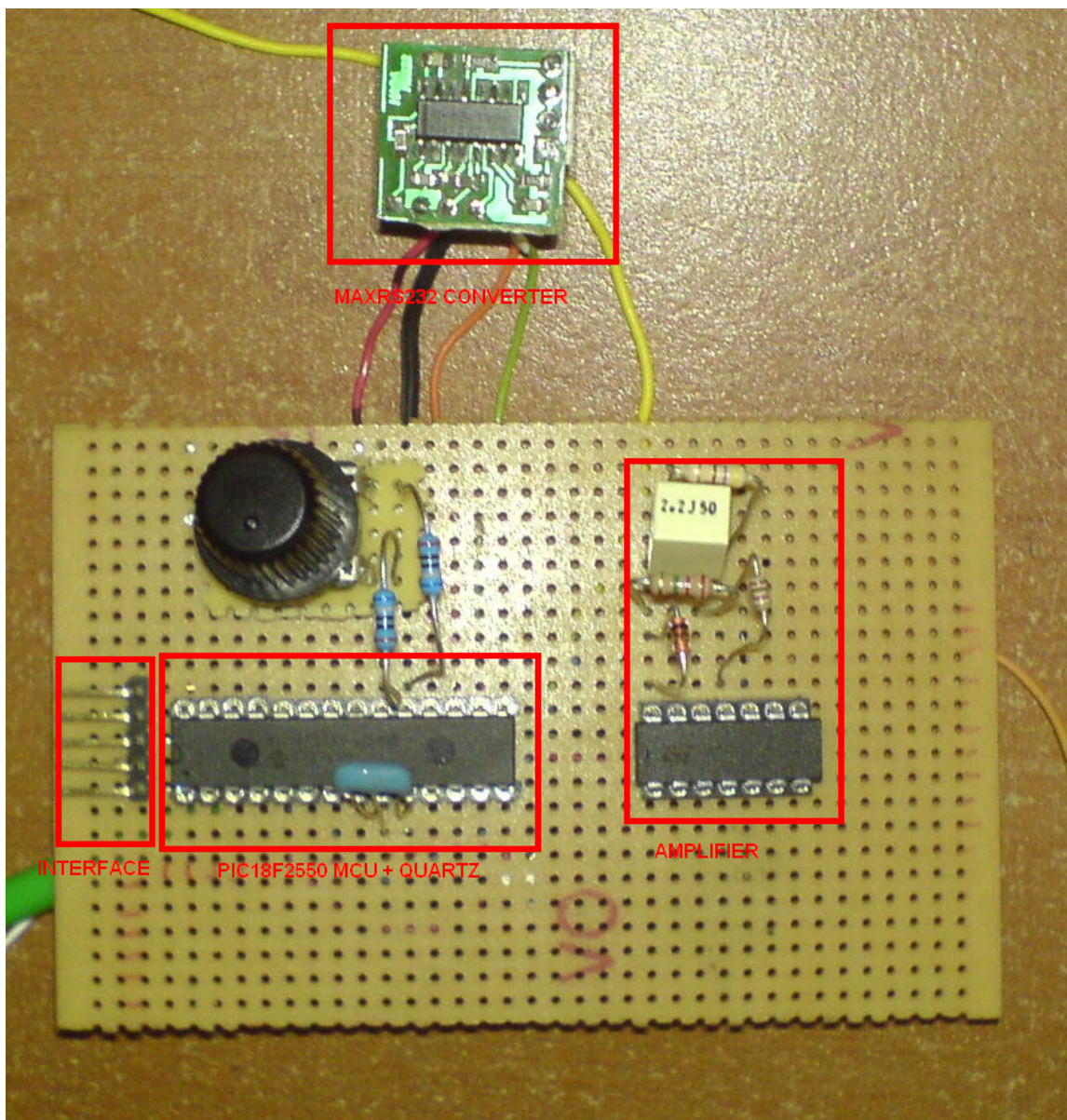
A tesztelés során úgy adódott, hogy az előerősítő áramkör jól követi a szenzoros bemenetet az összes rögzített jel esetén. Alább látható az előerősítő áramkör működése (**12.ábra**). A zöld jel az oszcilloszkópon a szenzor által generált jel (ami nem az eredeti, hanem a beépített jelgenerátor által van szimulálva). A kék jel az előerősítő kimenete. Látható, hogy a mV-os nagyságrendű jelből több voltos jel keletkezik, mely követi a pozitív csúcsokat, amelyek egy fogaskerék fogának szenzor előtt történő elhaladását jelzik.



12. ábra: Rögzített jel előerősítése működés közben

Az áramkör fizikai megvalósítása

Miután minden fontosabb adat rendelkezésre állt, elkészíthetővé vált az áramkör. Jelenleg egy próbapanelen foglal helyet a PIC mikrokontroller és az erősítő, illetve a későbbiekben felmerülő programozáshoz szükséges interface (13.ábra). Helyet kap még a panelen egy soros portos kommunikációhoz szükséges eszköz, a MAX RS232. Ennek szerepe annyi, hogy a mikrokontroller által előállított jeleket a PC által értelmezhető jelekké alakítja. Az átalakításra a tápfeszültségek különbözősége miatt van szükség.



13. ábra: A próbapanelen megvalósított áramkör fontosabb részegységei

PIC programozás

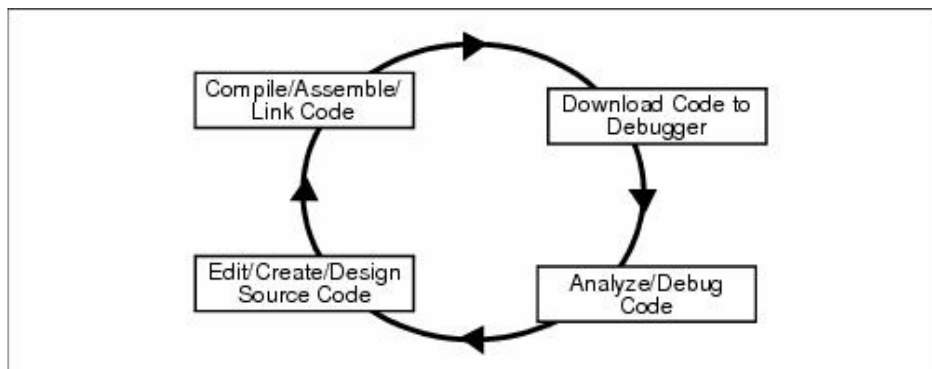
A fejlesztéshez használt környezet és eszközök

A következőkben a korábban már említett PIC mikrokontroller programozásához, a programok javításához és teszteléséhez elengedhetetlen alkalmazások rövid bemutatása, működésük alapszintű ismertetése következik

Mplab IDE

Windows alatt futó alkalmazás, melynek segítségével könnyedén fejleszthetők különböző alkalmazások a Microchip által gyártott mikrokontrollerekre. Integrated Development Environment-nek (integrált fejlesztői környezet) nevezik, mivel beágyazott rendszerek fejlesztéséhez szolgáltat megfelelő környezetet.

A beágyazott rendszerek nagyban különböznek a ma ismert PC-ktől. A lényeg, hogy egy dedikált, specifikus feladat megoldására a legalkalmasabbak, míg a PC-kkel általános és sokrétű feladatok (különböző programok futtatásával) oldhatók meg. Egy beágyazott mikrokontrolleren általában csak egy program fut. Ennek eredményeképpen olcsón előállíthatók a megoldandó feladat számítási igényeit figyelembe véve. Tehát az MPLAB IDE ilyen beágyazott rendszerek fejlesztésére alkalmas eszköz. A következő ábra mutatja a fejlesztés menetét (**14.ábra**). Általánosan elmondható, hogy első körben nagyon ritkán sikerül tökéletesen működő alkalmazást alkotni, ezért a fejlesztés jellege kvázi inkrementális. A kód megírása és lefordítása után a kapott kódot egy úgynevezett debuggolásra alkalmas eszköznek adjuk át, amiről a későbbiekben részletesen szólok. Az analizálás után visszatérhetünk hibáink kijavításához, a kód újraírásához.



14. ábra A fejlesztés körfolyamata(Design cycle)

Projekt Menedzser (Project Manager)

Az úgynevezett project manager segítségével könnyedén összerendezhetjük azokat a kódrészleteket, egyéb forrásokkal együtt, melyeket alkalmazásunkhoz fel szeretnénk használni. Ezek összerendezése után a linkerhez jutnak, melynek feladata ezek megfelelő elrendezése a programmemóriában, és megbizonyosodni arról, hogy minden egyes kapcsolódás úgymond feloldható-e.

Hibakereső (debugger)

Egy fejlesztői környezetben a keletkezett kód lefutása egy debugger segítségével tesztelhető. Ez a debugger lehet egy szoftver program, ami szimulálja a mikrokontroller működését tesztelési célból, vagy egy speciális eszköz, amely magát az eszközön futó alkalmazást elemzi. Az MPLAB IDE kétféle lehetőséget kínál ennek megvalósítására: a programozót (programmer) és a hardveres debuggert. A programozó szimplán beégeti a PC által készített gépi kódot a mikroprocesszor belső memóriájába. Ezek után csak reménykedhetünk abban, hogy a mikrokontroller a feladatot megfelelően látja el... A hardveres debuggolást egy úgynevezett In Circuit Debugger (ICD) végzi el.

Az MPLAB ICD viszonylag olcsó eszköz, amely mint a neve is mutatja egy áramkörön belüli hibakereső eszköz. Képes a mikrokontroller gyakorlatilag összes funkcióját az ellenőrzése alá vonni. Többek között a következő dolgok használatát teszi lehetővé.

- Realtime vagy lépesenkénti futtatás
- Breakpointok elhelyezése a programkódban, a mikrokontroller regisztereinek és változóinak megfigyelése és módosítása
- Áramkörön belüli hibakeresés
- Tápfeszültség figyelés
- Elemző LED-ek használata
- Az IDE használata
- RS232-es USB kapcsolat a PC-vel

A már korábban említett szoftveres szimulátor is használható debuggolás céljából. Ez az MPLAB SIM, amely szinte az összes PIC mikrokontroller család működését képes szimulálni

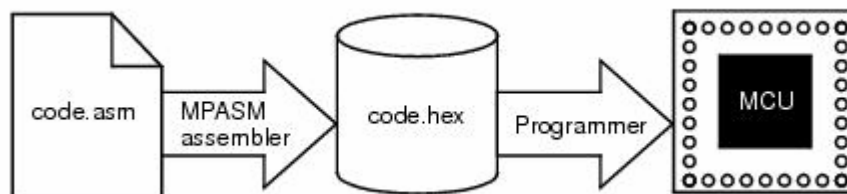
Az MPLAB IDE környezetbe van integrálva, és a következő szolgáltatásokat kínálja:

- A futtatható kód módosítása és azonnali újrafuttatása
- Külső jelek beillesztése a processzor működésének szimulációjába
- Közvetlen láb állapot és regiszter elérés

Az Assembler

Miért pont assembler

Az MPASM assembler egy parancssorból is futtatható Windows alapú PC alkalmazás, amely assembly nyelvű fejlesztői platformot szolgáltat a PIC-ek programozásához. Ez az assembler alapesetben egy abszolút kódot generál, amely egyből feltölthető a mikrokontrollerre (**15.ábra**).



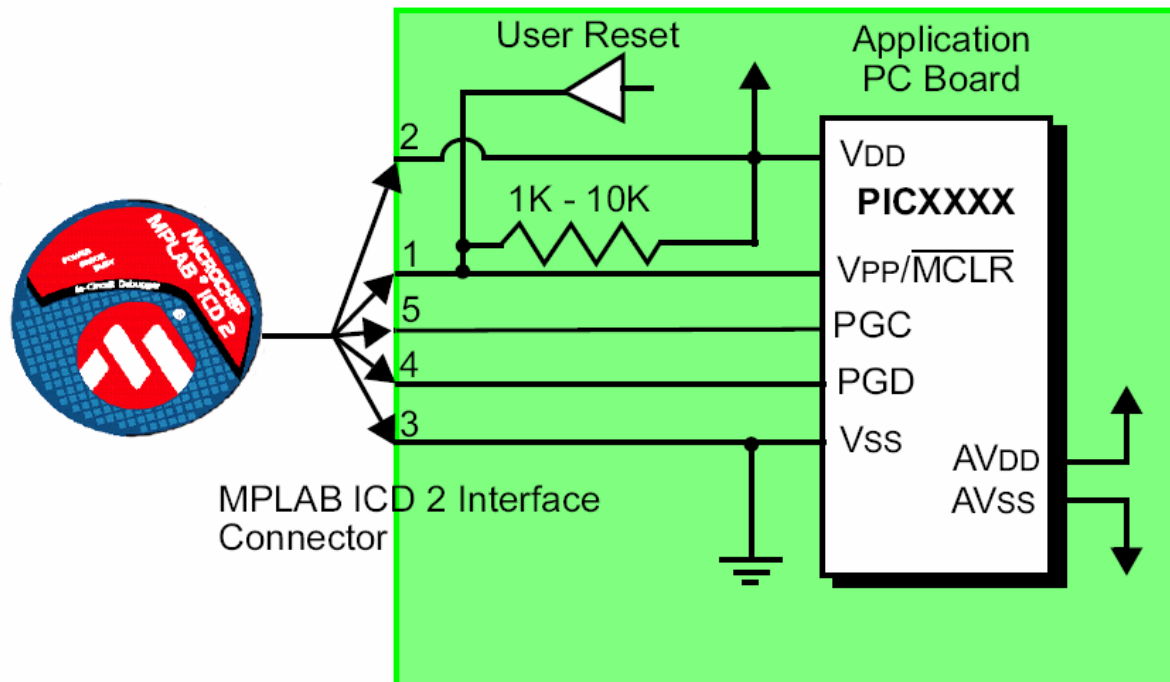
15. ábra: Az assembly kód állapotai

Az abszolút kód azt jelenti, hogy minden egyes elemének explicite definiálva kell lenni az összes kódrészlet és egyéb forrásfájlok között. Ha ez hiba nélkül megtörténik, egy hex fájl készül a feldolgozás végén, ami a futtatható gépi kódot tartalmazza. Ennek a kódnak a futását lehet aztán ellenőrizni egy debugger segítségével. Az általam fejlesztett kód is abszolút típusú.

A mikrokontroller csatlakoztatása

A fejlesztés mikrokontrolleren történő megkezdéséhez csatlakoztatnunk kell azt a hardveres debuggerhez, az Mplab ICD 2-höz. Ez a már korábban említett interface-en keresztül történik. A csatlakozás a mikrokontroller 5 lábának kivezetéséből áll, melyek egy UTP kábel segítségével köthetők az ICD-re. A mikrokontroller két I/O

portját, a tápfeszültséget, a földet és a reset lábát kell a debug eszköznek a helyes működés érdekében elérnie [14]. A pontos bekötés alább megfigyelhető (16.ábra)



16. ábra: Az ICD és a mikrokontroller csatlakozó felülete

Ha az ICD érzékeli a helyes bekötést, megkezdődhet a fejlesztés a PC-n keresztül. Ez az interface látható a 16. ábrán, melyhez a csatlakozót magam készítettem.

A feldolgozás elméleti alapjai

A hajtókar szögpozíciójának megállapítása – ismételten – a következő alapokon nyugszik. Számolni kell, hogy a referencia fog észlelése óta hányadik fogat érzékeltük, illetve a referencia fog elhaladását is észlelni kell. Ebből a két alapvető feladatból meglehetősen sok további feldolgozási lépés következik.

A fogaskerék forgása során a szenzor által generált és felerősített jeleket a mikrokontroller egyik bemenetére adom, hogy az a megfelelő módon feldolgozhassa. Itt azonban végig kell gondolni néhány fontos tényezőt. Milyen tulajdonságú lehet egyáltalán a jel, mit kell megállapítani ebből a jelből, hogyan lehet megállapítani azt, ahhoz hogy a már korábban említett referenciafoghoz viszonyított fogszámlálás alkalmas legyen a pedálszög megállapítására?

Az első fontos dolog, hogy tudjuk hány darab fog van az adott fogaskeréken. Az általam elérhető tricikli fogaskerekén összesen 48 található. Ez $7,5^\circ$ -os felbontást tesz lehetővé, amely a tanulmányozott ingerlési mintázatok előállításához tökéletesen elegendő.

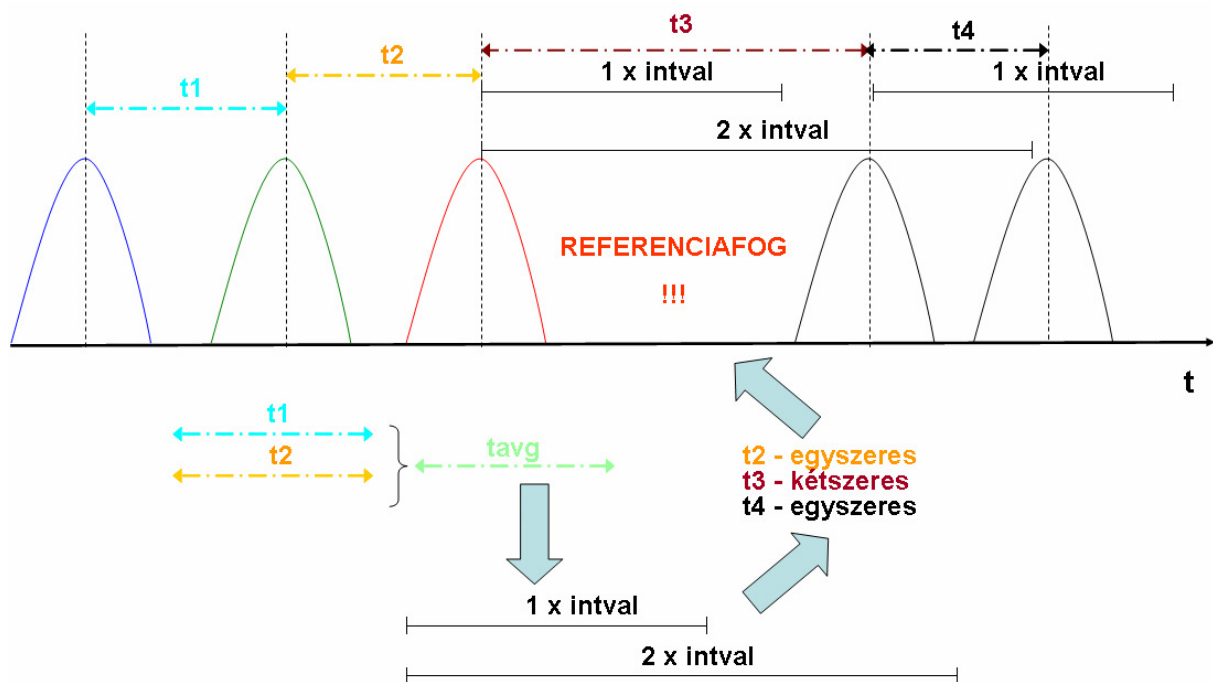
Az előzetes mérések során meghatároztam, hogy egy lassú tekerés meglehetősen alacsony frekvenciájú jelet ad. Ez fizikailag lefordítva annyit jelent, hogy a fogaskerék fogai viszonylag ritkán haladnak el a szenzor előtt. Másrészt gyors tekerésnél nagyobb a frekvencia. A biztonságos működés szempontjából és az egyébként is adott fizikai korlátok miatt szükséges feltételeznem a bemenő jelről, hogy 3Hz és 200Hz közötti frekvenciatartományban lehet. Ez azt jelenti, hogy 1s alatt minimum $22,5^\circ$ fokos maximum 1500° fokos (kicsivel több mint négy fordulat) pedálszög változás lesz mérhető. A határokon kívül eső jel esetén nem lesz mérhető a hajtókar szögállása. Ez az elérhető mérési tartomány.

A mikrokontroller meglehetősen diszkrét eszköz a 8 bites regiszterei miatt. Ezért is van szükség ilyen korlátok meghatározására, hogy a későbbiekben reprezentálhatók és érzékelhetők legyenek a bejövő információk. Másrészt, ha nem tudnám határok közé szorítani a bemenő jelet és annak dinamikáját, akkor valamiféle adaptív módszerrel kéne szabályozni a feldolgozást, amely meglehetősen elbonyolítaná a helyzetet.

Tehát a feltételezett határok megléte miatt megadható a maximális és minimális gyorsulás illetve lassulás is, melyek dinamikai feltételeket szabnak a jelnek. Ezek százalékos rendszer-konstansokként így leírhatók a későbbiekben, számításukat a gyakorlati megvalósítás leírásakor a korlátvizsgálatnál részletesen elemzem.

A tekerés dinamikájának ilyen ismeretére azért van szükség, hogy megállapíthassam, mikor lehet egy adott fog jelének érkezésére számítani. Ennek ismerete elengedhetetlen a referencia fog felismerése miatt, hiszen itt pont az jelenti az információt, hogy nincs jel. Bővebben kifejtve, ha ismerem a tekerés aktuális dinamikáját és a következő jel vélt érkezésekor nem történik meg a jel vétele – a határok által megszabott időn belül –, viszont az azutáni jel vélt érkezése után pontosan ugyanolyan dinamikával folytatódik a tekerés, mint korábban, akkor biztosan állítható, hogy a referenciafog „haladt el” a szenzor előtt. Ekkor újra kell kezdeni a fogak számlálását és így a szöghelyzet jelzését. Ezek alapján a feldolgozáshoz szükség lesz a fogak által gerjesztett impulzusok között eltelt idő mérésére és ezen adatok tárolására. A jelek között partikulárisan eltelt időről nehéz

lenne feltételezni, hogy pontosan megadható, ezért egy aktuális – az éppen elhaladó fog szempontjából lokális – átlag számításával viszont könnyebben tesztelhető a fenti dinamikai feltételek teljesülése. A folyamat szemléletes ábrázolása itt látható (17.ábra) A dinamikai feltételt azok az intervallumok jelentik, amelyek az átlagosan eltelt időből kerülnek meghatározásra.



17. ábra

A mikrokontroller felhasznált egységei

A dolgozat ebben a részében ismertetem a feladat megoldásához felhasznált integrált áramköri egységek működését és egy - két fontosabb PIC jellemzőt.

A mikrokontroller helyes működéséhez elengedhetetlen a globális tényezők beállítása és a megfelelő egységek konfigurálása. Ennek hiányában, jobb esetben el sem indul, rosszabb esetben nem a kívánt módon működik. Ezek kezelésére két lehetőség van. A mikrokontroller globális, a működést szabályozó elemeinek beállítását úgynevezett konfigurációs (config) regiszterek beállításával lehet elvégezni, míg az egyes részegységek működését vezérlő (control) regiszterek segítségével lehet beállítani.

Konfigurációs regiszterek

A konfigurációs regiszterekben levő bitek beállításával a mikrokontroller különböző eszközeinek alapvető működési módjai közül választhatók ki a megfelelők. A következő táblázat mutatja az elérhető regisztereket (**18.ábra**). Az első két oszlop a regiszter memóriacímét és nevét majd rendre a többi a beállítható bitjeit, végül az utolsó oszlop az alapbeállítást.

File Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Default/ Unprogrammed Value
300000h	CONFIG1L	—	—	USBDIV	CPUDIV1	CPUDIV0	PLLDIV2	PLLDIV1	PLLDIV0	--00 0000
300001h	CONFIG1H	IESO	FCMEN	—	—	FOSC3	FOSC2	FOSC1	FOSC0	00-- 0101
300002h	CONFIG2L	—	—	VREGEN	BORV1	BORV0	BOREN1	BOREN0	PWRTE \bar{N}	--01 1111
300003h	CONFIG2H	—	—	—	WDTPS3	WDTPS2	WDTPS1	WDTPS0	WDTEN	---1 1111
300005h	CONFIG3H	MCLRE	—	—	—	—	LPT1OSC	PBADEN	CCP2MX	1--- -011
300006h	CONFIG4L	DEBUG	XINST	ICPRT ⁽³⁾	—	—	LVP	—	STVREN	100- -1-1
300008h	CONFIG5L	—	—	—	—	CP3 ⁽¹⁾	CP2	CP1	CP0	---- 1111
300009h	CONFIG5H	CPD	CPB	—	—	—	—	—	—	11-- ----
30000Ah	CONFIG6L	—	—	—	—	WRT3 ⁽¹⁾	WRT2	WRT1	WRT0	---- 1111
30000Bh	CONFIG6H	WRWD	WRWB	WRWC	—	—	—	—	—	111- ----
30000Ch	CONFIG7L	—	—	—	—	EBTR3 ⁽¹⁾	EBTR2	EBTR1	EBTR0	---- 1111
30000Dh	CONFIG7H	—	EBTRB	—	—	—	—	—	—	-1-- ----
3FFFFEh	DEVID1	DEV2	DEV1	DEV0	REV4	REV3	REV2	REV1	REV0	xxxx xxxx ⁽²⁾
3FFFFFh	DEVID2	DEV10	DEV9	DEV8	DEV7	DEV6	DEV5	DEV4	DEV3	0001 0010 ⁽²⁾

18. ábra: A PIC18F2550 konfigurációs regiszterei, memóriacímeikkel és alapértelmezett értékeikkel

A legfontosabb az első két regiszter, melyek segítségével a PIC oszcillátorát tudjuk beállítani. Különböző órajel osztásokat az első regiszterrel és különböző oszcillátor típusokat a második regiszterrel. A többi regiszterrel vezérelhetők a mikroelektronikából jól ismert áramköri jellemzők és egységek, mint a Watch Dog Timer(WDT) Power Up Timer, Brown Out Reset, reset engedélyezés illetve memóriaterületek írásvédettségének kezelése. Itt állítható be az ICD működésének engedélyezése illetve a kibővített utasításkészlet alkalmazása.

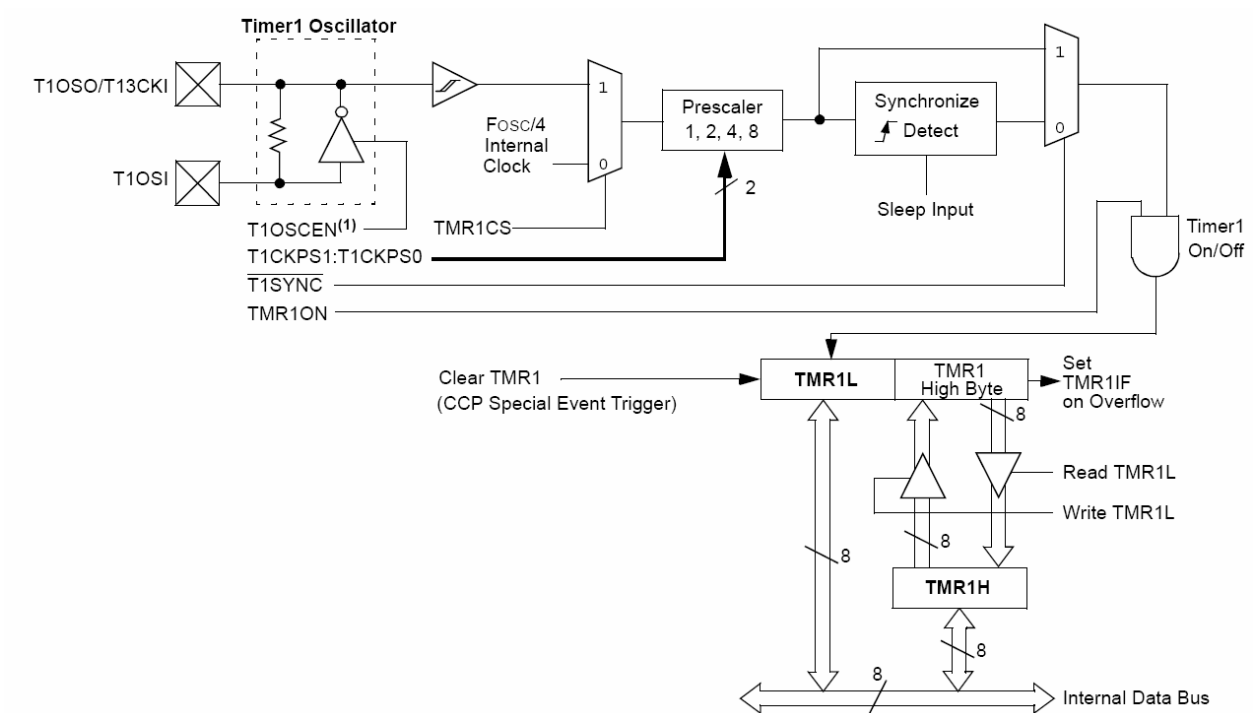
Timer

Az előző fejezetből is látszik, hogy az időmérés igen fontos tényező, ugyanis a szögsebesség illetve abszolút szögpozíció meghatározásának alapvető eszköze a

feladat megvalósításában. Ezt a funkciót a mikrokontrollerbe épített timer/counter egység felhasználásával lehet garantálni, melynek jellemzői a következők.

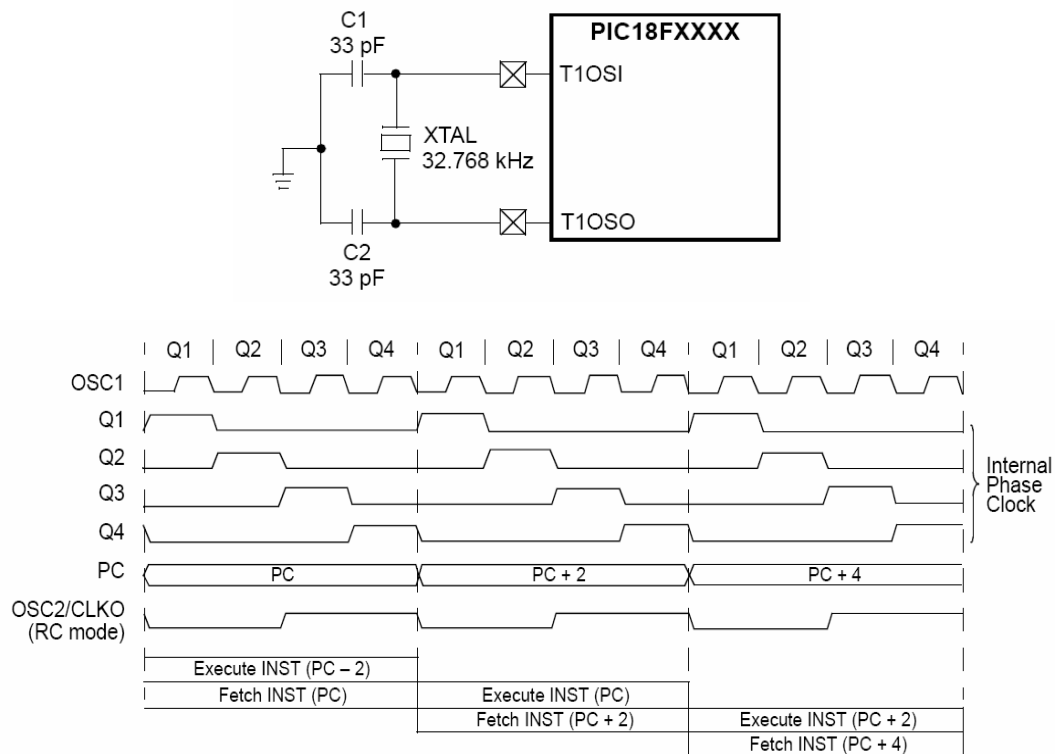
- Szoftverből állítható 16 bites időmérő vagy számláló funkció
- Írható olvasható 8 bites regiszterek (TMR1H és TMR1L)
- Az időalap forrása kiválaszthatóan külső vagy belső
- Túlcscordulás jelzése (interrupt ld. később)
- Órajel státuszát-jelző flag (T1RUN)

A timer egység működését bemutató blokkdiagram (19.ábra)



19. ábra A Timer1 egység felépítése

Látható, hogy kiválasztható a meghajtó oszcillátor típusa, amely lehet belső vagy külső, illetve a fontosabb regiszterek a TMR1H és TMR1L, amelyekben az idő folyamatosan inkrementálódik az oszcillátor frekvencia függvényében. A TMR1ON bittel indítható. Megfigyelhető, hogy egy logika segítségével konzisztens időadatot tudunk kiolvasni a regiszterekből, mert ha olvassuk a TMR1L-t, akkor a felső bájt tartalma a TMR1H buffer regiszterbe másolódik, míg az eredeti számláló rég ketyeg tovább. Az általam megvalósított rendszerben a pontosság miatt külső kvarc oszcillátort használok, amelyet a következő módon kell a mikrokontrollerhez illeszteni (20.ábra).



20. ábra: Kvarc illesztése a mikrokontrollerhez és az általa meghatározott oszcillátor frekvencia felosztása az utasítás-végrehajtás során

Az ábrával ellentétben az általam használt kvarc 8Mhz-en oszcillál. Ebből meg lehet állapítani egy utasítás végrehajtásához szükséges időt, melyből egyenesen következik a timer egy inkrementálódásának ideje.

Ezt az egységet is vezérlő regiszter segítségével lehet beállítani, melynek neve TMR1CON.

Meghatározott időintervallum mérésére könnyedén konfigurálható, ha írjuk a regisztereit és tudjuk az utasítás-végrehajtás ciklusidejét. Esetemben pontosan 1ms mérésére használok fel ezt az egységet, megszakítás-vezérléses üzemmódban (azaz ha eltelik a megfelelő idő interrupt keletkezik). A pontos konfiguráció a gyakorlati megvalósítások leírásánál szerepel.

Interrupt kezelés

Mire használunk IT-t a rendszerben

Az interrupt egy program futtatásával aszinkronban bekövetkező jelenség, mely a programmenet megváltozásának szükségét jelzi. Ez nem egy kimondott egysége a mikrokontrollereknek, de egy nagyon hasznos funkció, mely nélkül elképzelhetetlen

lenne a működés. Pontosítva, ha nem lehetne megszakítani a vezérlést, akkor csak egy többnyire lineáris alkalmazás készülhetne el. Azért van szükség ebben az alkalmazásban interruptokra, mert bizonyos események bekövetkezésekor azonnal, a megfelelő módon kell reagálni. Ez fontos mind a pontosság, mind a valós idejű működés szempontjából. Ilyen esemény, amikor a szenzoros jel egy fog elhaladásakor megváltozik. Ez egy olyan esemény, amit kezelni kell amint lehet, azaz a megfelelő utasításokat végre kell hajtani akárhol is tartson a vezérlés. Hangsúlyos, hogy meghatározott események bekövetkezésekor kérhető a futás megszakítása.

A PIC18F2550 megszakításai (IT) több forrásból is származhatnak, több esemény is kiválthatja őket, ráadásul prioritással is elláthatók (az én alkalmazásomban nem szerepelnek prioritással rendelkező IT-k). Az úgynevezett interrupt vektor a memória elején helyezkedik el a 000008h hexa számmal jelölt memóriacímen. Ez azért fontos, mert ha bármilyen IT megszakítja a program futását a vezérlés azonnal erről a címről folytatódik tovább, bár ehhez bizonyos feltételeknek teljesülni kell. 10 vezérlő regiszter segítségével konfigurálhatjuk a megszakítás vezérlést, ezek a következők: RCON, INTCON1-3, PIR1-2, PIE1-2, IPR1-2. Minden kiváltható IT működése 3 bit beállításával adható meg.

Egy jelző bit mely az IT keletkezését jelzi, egy engedélyező bit, mely lehetővé teszi a futás vezérlésének megszakítását és a kezdeti interrupt vektorra ugrását, ha a jelző bit bebillent. A harmadik a prioritások jelzésére szolgál. Ha megtörténik az IT akkor egy globális engedélyező bit is vált, hogy ne következhesen még egy. A végrehajtás aktuális címe eltárolódik, s egy időben megkezdődik az IT kiszolgáló rutin, mely során le lehet kérdezni a különböző jelző biteket, hogy melyik okozta az IT-t. A visszatérés az IT-ből a *retfie* nevű utasítás elérésével történik, ekkor az eltárolt utasítás memóriacímétől folytatódik a programfutás. A vezérlő regiszterekben ezeket a biteket lehet beállítani és szabályozni, hogy mikor milyen IT válthódhasson ki.

Fontos megjegyezni, hogy nem csak a mikrokontroller belső egységei válthatnak ki IT-t, hanem valamely külső jel is lehetővé teszi ezt. Pontosán ez a lehetőség fontos a pedálszög feldolgozásának megkezdésében. Az előerősített jel megfelelő arra, hogy kiváltson egy ilyen IT-t, és utána meghatározhassuk, hogy mi történjék a feldolgozás során. Ilyen interruptot kiválthat a PIC egy lábán történő jelszintváltozás, illetve maga a port állapotának változása.

Két ilyen lehetőség van: egyik az úgynevezett INTx interrupt, amely egy adott (x) porton figyel és felszálló élre vagy leszálló élre vált ki IT-t. A másik a PORTB interrupt-on-change, amely szintén megfelelő lábon bekövetkező jelszint változás esetén vált ki IT-t.

Ez a funkció mindamelllett, hogy segít az egyes belső egységek eseményeinek kezelésében, gyakorlatilag megteremti a pontos és valós idejű kapcsolatot a külvilág – szenzoros információ – és a mikrokontroller belső világa között.

Soros kommunikáció

Miért is kell ez nekünk?

A soros porton történő kommunikáció elterjedt metódus adatok küldésére egy számítógép és egy perifériás eszköz között, mint például egy programozható mikrokontroller. Az adatok továbbítása egyetlen vonalon történik a vevő felé, és egy időben csak egy bitnyi információ kerül elküldésre. Használata előnyös, ha nem kell nagy mennyiségű adatot küldeni, és eléggé elterjedt mivel majdnem minden PC rendelkezik egy vagy több soros porttal. A kommunikáció megvalósításához nem szükséges semmilyen plusz hardver egy darab kábelen kívül.

Alkalmazásához a következő négy paramétert kell meghatározni:

- Az átvitel baud-rate-je
- Egy karakter kódolásához felhasznált bitek száma
- A paritásbit érzékelése (használatát opcionális)
- Stop bitek száma

Minden egyes elküldött karakter egy keretbe van ágyazva (frame), melynek részei a startbit, őt követik az adatbitek, majd az opcionális paritásbit, és végül a stop bit (vagy bitek)

A baud rate annak a mértékegysége, hogy két soros porton kommunikáló eszköz között milyen gyorsan mozognak az adatok.

Az RS-232 standard soros portos kommunikációban két feszültségszint (egy magas és egy alacsony) lehetséges. Ilyen kétállapotú kódolás esetén a baud rate megegyezik a csatornán másodpercenként küldött információ maximális bitszámával, beleértve a vezérlő biteket is.

Ha a jelszint $> 3V$, akkor az adat bitértéke 0, ha jelszint $> -3V$ akkor 1.

A soros kommunikációban a kimeneti jel feszültsége általában +12 V és -12 V körül van. A 3 és -3 V közötti rés a zaj elnyelését teszi lehetővé.

A start bit – egy felszálló él – jelzi minden egyes frame kezdetét, melynek hossza másodpercben a baud rate reciprokával egyenlő

9600 baud rates átvitelt feltételezve a start bit hossza (és az összes többi is) megközelítőleg 0.104 ms, és egy 11 bitből álló karakter frame átviteléhez körülbelül 1.146 ms-ra van szükség.

Az adat bitek negáltan és fordított irányban kerülnek átvitelre, ami azt jelenti, hogy az LSB-től az MSB-ig küldjük őket, és 1-nek értelmezzük a negatív feszültséget és 0-nak a pozitívot.

Az adat biteket egy opcionális paritás ellenőrző bit követheti egy frame-ben. Egyszerű hibaellenőrző szerepet játszik azzal, hogy – az átvitel kezdetekor meghatározott – páros vagy páratlan számú egyesre egészíti ki a frame-t.

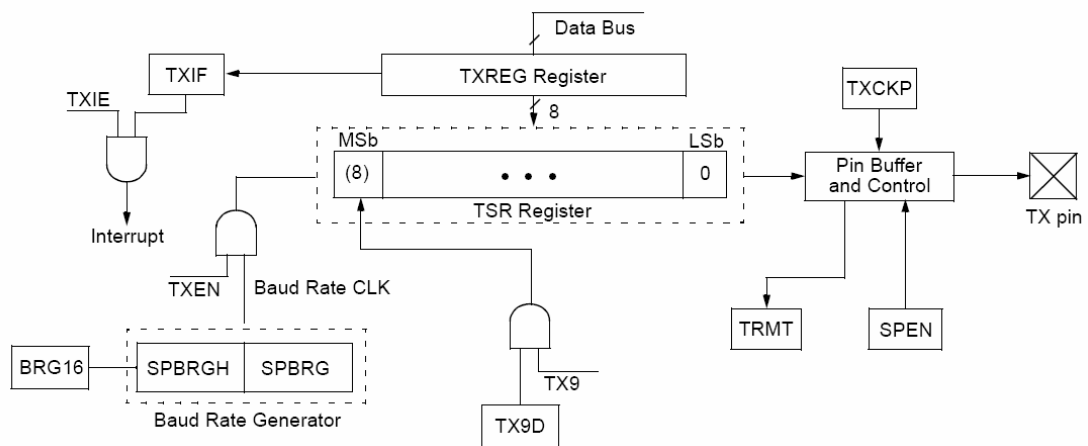
A karakter keret utolsó része egy 1, 1.5, vagy 2 bit hosszúságú stop jel. Ezek mindig negatív feszültségűek. Ha nincs több átvitelre szánt karakter az átviteli csatornán, akkor annak feszültsége továbbra is negatív marad, míg nem érkezik újabb adat, amit mindenképpen pozitív feszültség jelez.

A PIC18F2550 egyik soros kommunikációra képes modulja az EUSART (Enhanced Universal Synchronous Asynchronous Receiver Transmitter). Mint a neve is mutatja szinkron és aszinkron módban is működtethető half-duplex és full-duplex üzemmódban is. Speciális jellemzője még, hogy automatikusan felismeri az adatátvitel baud-rate-jét, illetve speciális break (megszakító) karaktert (12 bit hosszú) is tud küldeni. TRISC és SPEN regiszterek speciális bitjeinek beállításával engedélyezhetjük használatát. TXSTA RXSTA BAUDCON vezérlő regisztereken keresztül szabályozhatjuk az átvitel tulajdonságait.

Az egység beépített baud-rate generátora gondoskodik a helyes “ütemezésről”, ennek mikéntjét nem részletezem. Kiválasztható az átvitel kívánt tulajdonságaihoz a megfelelő beállítás egy táblázatból.[15] A feladat megvalósítása szempontjából mindenképpen aszinkron kommunikációt kell használni a kontrolleren. Ez könnyen kivitelezhető a fent említett regiszterek figyelmes beállításával.

Az átvitel lelke a Transmit Shift Register (TSR), mely egy írható olvasható bufferből (TXREG) nyeri a tartalmat. Ezt a regisztert lehet a programból írni. Amikor ez megtörtént, a tartalma átkerül a TSR-be, amint az előző tartalmának stopbitjét is továbbította. Ezek után az adott beállítások által adott módon a TSR úgymond

kishifteli a TX lábra a tartalmát. Amint ez megtörtént egy bit (TRMT) jelzi, h üres a TSR. A kommunikáció működésének blokkvázlata alább látható (21.ábra)



21. ábra: A soros kommunikációért felelős egység felépítése

Adat címzési módok

A PIC Harvard architektúrájú, ami annyit jelent, hogy külön helyen tárolódnak a memóriában az adatok és a programok. Míg a programmemóriában levő címeket csak a Program Counter (PC) nevű speciális regiszteren keresztül lehet elérni, az adatmemóriában levő adatok többféle módon címezhetőek. Ezek a következők:

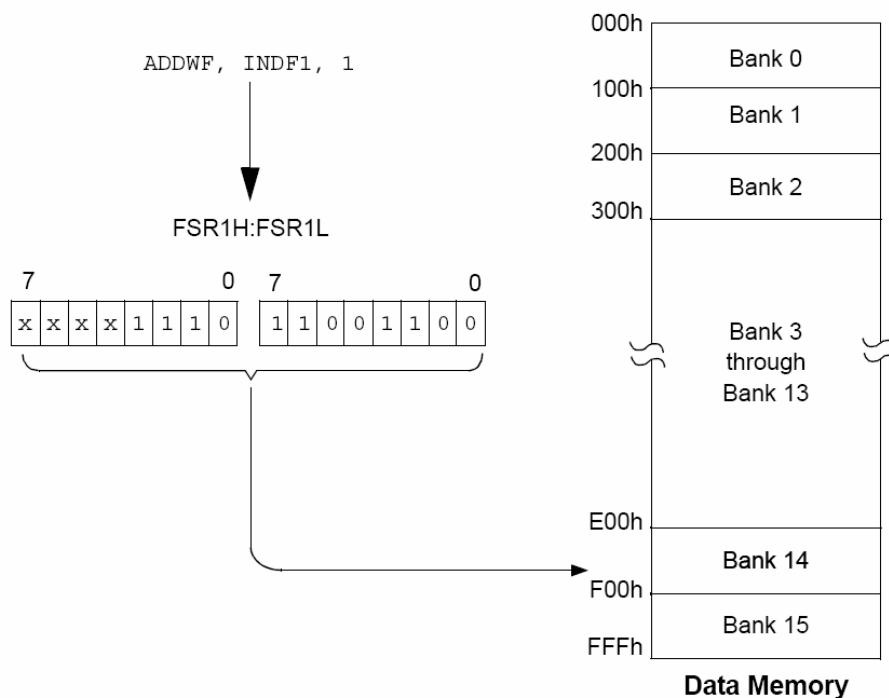
- Inherens
- Literál
- Direkt
- Indirekt

Inherens címzés esetén nincs szükség semmilyen paraméter megadására, az utasítás globális hatással van a mikrokontrollerre (ahogy a neve is mutatja, együtt jár az utasítással), vagy implicite módosítja egy regiszter tartalmát (pl. RESET SLEEP). A literál címzés, mint a neve is mutatja lexikális típusú adatok megadását kívánja, és ezek memóriába történő bevitelére szolgál (pl. MOVLW ADDLW)

A direkt címzés megadja a forrás és célmemória címét is az utasítás után, illetve felkínálja többek között azt az opciót, hogy az utasítás eredménye hol tárolódjon.(MOVFF).

Indirekt címzés alkalmazásával elérhetők az adatmemóriában levő tárolt adatok fix elérési cím megadása nélkül. Ez úgynevezett File Select Register-ek (FSR) használatával valósulhat meg, melyek egyfajta mutatóként (pointer) működnek. Meghatározzák a kívánt helyet ahova írni, vagy ahonnan olvasni szeretnénk. Mivel az FSR-ek is az adat memóriában tárolódnak közvetlenül manipulálhatók egy programon belül. Ez teszi nagyon hasznossá őket, segítségükkel különböző adatstruktúrák implementálása válik lehetővé.

Három ilyen FSR érhető el a PIC-en belül. Mindegyik 2 darab egyenként 8 bites regiszter. Ezek mindegyike egyfajta lineáris leképezései a rendelkezésre álló memóriának. Tehát a tartalmuktól függően más és más memóriacímek címezhetők meg. Az elérésük úgynevezett virtuális regiszterek olvasásával illetve írásával történik. A következő ábra ezt a funkciót mutatja (**22.ábra**).



22. ábra: Az adatmemória indirekt címzési módja

Az `ADDWF, INDF1, 1` utasítás a `W` regiszter tartalmát adná az `INDF1` regiszter tartalmához. Ez azonban egy speciális virtuális regiszter, melynek címzésekor az `FSR`-ek által mutatott memóriaterület címződik meg (**22.ábra**). Többféle speciális virtuális regiszter címzésével is elérhetjük az `FSR`-ek által mutatott címek tartalmát. Ezek a következők:

- `INDFx` – megcímeződik az `FSRx` által mutatott memóriaterület ($x=0..2$)

- POSTINCX – megcímeződik az FSRx által mutatott memóriaterület ($x=0..2$) és miután a művelet végrehajtott az FSR által mutatott cím eggyel inkrementálódik
- POSTDECX – megcímeződik az FSRx által mutatott memóriaterület ($x=0..2$) és miután a művelet végrehajtott az FSR által mutatott cím eggyel dekrementálódik
- PREINC – az FSR által mutatott cím eggyel inkrementálódik, majd csak ezután címeződik meg az FSRx által mutatott memóriaterület

Gyakorlati megvalósítás

Az eddigiekben bemutattam azokat az egységeket és funkciókat, amiket megismertem és felhasználtam a feladat megoldása során. Most elemzem az egyes részfeladatokat és ezek megvalósításait, amiknek összeállításával megvalósult a kitűzött célt megvalósító program.

Az ismertetés során a könnyebb érthetőség kedvéért nem foglalkozom teljes mélységében az elemi adatok reprezentálásával (bájt – 2 bájt konverzió), csak ahol kiemelt szerepe van. A teljes programkód a mellékletben található.

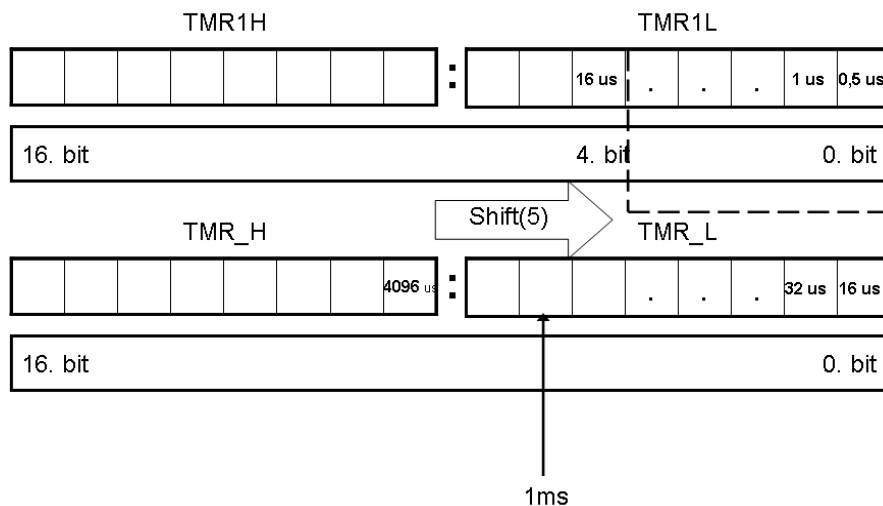
Időmérés a timerrel

Az idő mérésének célja, hogy pontosan megállapítható legyen, mikor történnek meg a várt események. Ehhez biztosítani kell egy időalapot ami alapján lehet mérni az időt. Ezt adja a kvarc rezgése, amely közvetlenül vezérli a timer regiszter működését. Esetemben 8 MHz-es kvarc használatával a timer, indítása után 2 MHz-es frekvenciával inkrementálódik. A mérésre kitűzött időalap 1 ms., mivel a feladat nagyságrendje ezt követeli meg. A 2 MHz-es frekvencia 0,5 us-onkénti inkrementációt jelent, miszerint pontosan 2000 növekedés eredményez 1ms-os időalapot. Beállítva ezt az értéket a regiszterbe meg is van a kívánt alap. Mivel a timer 16 biten működik (8 bit kevés lenne) hexa 0F830h-at írva a regiszterbe pontosan 1ms-elteltével az indítástól IT keletkezik. Ekkor újra be kell állítani a regiszterbe a megfelelő értéket és újraindítani az órát. Egyetlen dologra kell figyelni

még: a valós idő akkor is telik, amikor a timer újra inicializálódik. Ezért az inicializálásban levő parancsok számával még meg kell növelni a regiszter tartalmát.

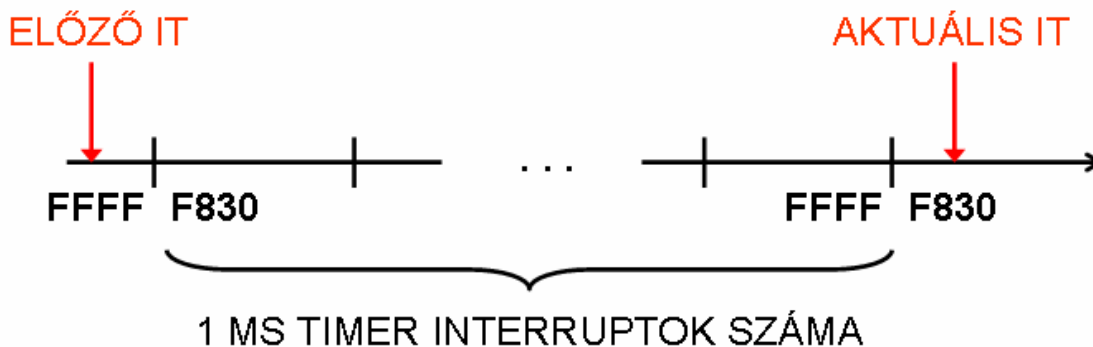
Adatreprezentáció

A meglévő időalap jó kiindulópont, amikor két fog elhaladása között eltelt időt kell meghatározni, azonban korántsem elegendő. Ahhoz, hogy értelmes idő információt lehessen megadni az egyes események bekövetkezéséhez, szűkíteni kell a lehetséges időértékek halmazát, mivel a digitális formában kell azt reprezentálni. Felsőbb körökben ezt hívják típusérték-halmaznak. A mérések során úgy adódott, hogy 200Hz-nél nagyobb frekvenciával nem jöhet szenzoros jel. Saját feltétel, hogy 3Hz-nél kisebb frekvenciával történő tekerés nem tekinthető értelmezettnek. Ezekből a megszorításokból következik a maximális illetve minimális pontossági igény, azaz mennyire akkurátusan kell tárolni az eltelt időt. 200 Hz (intervallum alsó határa) és 3Hz (intervallum felső határa) között. 1%-os hibátűrést feltételezve a 200 Hz-es tekeréshez tartozó 5 ms-ot meg kell különböztetni 4,95 ms-tól. Tehát a megfelelő pontosság eléréséhez (feltételezve az adott hibátűrést) 50 us pontosság a mérvadó. Megfelelő biztonsági margót hagyva, a timer alsó 5 bitjét ki lehet hagyni a reprezentációból (mivel a legalsó bit 0,5 us-onként váltakozik, a megmaradó 6. bit 16 us-mal fog, ami jóval rövidebb(így pontosabb) mint az elvárt 50 us). A felső határt a 3Hz-ből adódóan a 333ms adja, amely 16 us-ot jelölő alsó bittől számolva minimum 9 bites ábrázolást feltételez. Mivel a PIC 8 bites architektúra még egy bájtot fel kell használni a mérendő idő reprezentálására. A következő ábra szemlélteti ezt a műveletet (**23.ábra**).



23. ábra: Az mért idő reprezentációjának kialakulása, és a tároló regiszter

Így alakul az idő tárolására szükséges memóriaterület. Ezzel azonban még csak az időpillanatok eltárolására adódik lehetőség. Ahhoz, hogy a megfelelő idődifferenciák adódjanak, két időbélyeg különbségét kell venni, amelyeket a fogak indukáltak. A probléma ezzel az, hogy nem tudható előre mikor történik meg a várt esemény. A következő ábra szemlélteti, hogy hogyan alakul ki a ténylegesen tárolt eltelt idő (24. ábra)



$$\text{ELTELT IDŐ} = [\text{FFFF} - \text{E_IT}] + [\text{A_IT} - \text{F830}] + [\text{TMR_IT} - 1][\text{MS}]$$

24. ábra: Az eltelt időt tároló regiszter végleges tartalmának alakulása

A fenti ábrán a műveletek nem igazi összeadásként értendők, a megfelelő helyre írva az eltelt időt tároló regiszterbe kapunk megfelelő eredményt. A fentebb részletezett bit-eltolás az összeg megléte után is végezhető. Az eddigiekben említett feladatok a szükséges konfigurációs regiszter beállításokkal együtt az IT.asm fájlban található, mely a melléklet része.

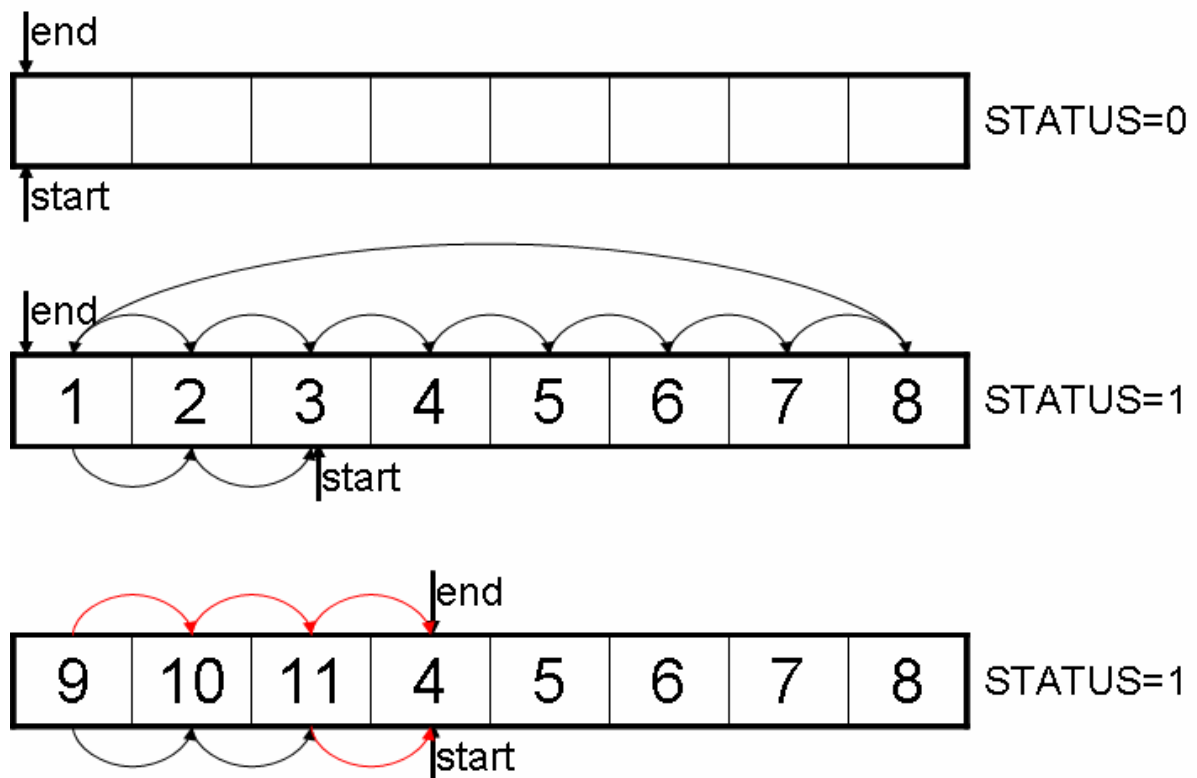
Gyűrűs regiszter

Az előbbieken bemutattam, az eltelt idő reprezentációjának alakulását. Ez a rész az így kialakult 2 bájtos regiszter tartalmának a hosszabb távú (egy bizonyos ideig tartó) eltárolásáról szól. Erre a feldolgozás elméleti alapjainak részletes leírásában található módszer miatt van szükség. Nevezetesen, az aktuális eltelt időt össze kell hasonlítani az azelőtt eltelt idők átlagával, hogy meghatározhatóvá váljanak a dinamikára vonatkozó korlátok.

Ennek kivitelezésére szükség van egy véges méretű adatstruktúrára, melynek segítségével könnyen elérhetők a szükséges adatok, illetve esetleges rajtuk végzett műveletek elvégzésére is lehetőséget ad.

Az én választásom az adatok tárolására egy megadható méretű gyűrűs regiszter (ring buffer, circular buffer), melynek eleje és vége kapcsolódik, így téve lehetővé az érkező adatfolyam megfelelő ideig történő tárolását. Ez azt jelenti, hogy a memóriában az utolsó memóriacímre történő írás után automatikusan az első címre történik az írás, az esetlegesen már benn lévő adatok egyszerűen felülíródnak. Ebből következik, hogy a buffer hossza implicite meghatározza azt az időintervallumot, amelyben az érkezett jelek relevánsak és az átlag számításában szerepelnek.

A fizikai megvalósítása meglehetősen sokféleképpen lehetséges, de szükséges – mint általában minden adatstruktúra kódolásának – feltétele a memóriaponterek használata. Az én megvalósításomban a buffer rendelkezik egy kezdő- és egy végcímet jelző pointerrel. A buffer írásakor a végcímmre mutató pointer nő, kiolvasáskor a kezdőcímmre mutató. Amikor megtelik a megadott hossz, és a végcím utoléri a kezdőcímet, a végcímmre mutató pointer inkrementálódásával a kezdőcímen levő mutató is inkrementálódik (azaz a pointer „tolja maga előtt” a másikat) (**25.ábra**). Erre azért van szükség, hogy az aktuálisan érkezett adatok mindig a legrégebbit írják felül, és ne történhessen meg, hogy elveszzenek az eddig tárolt adatok.



25. ábra: A ring buffer pontereinek működése és az állapotot jelző STATUS regiszter

A működés még így sem teljesen egyértelmű. Az okoz problémát, amikor a kezdő- és a végcímre mutató pointer ugyanarra a memóriacímre mutat. Ekkor tele van a buffer vagy üres? Ennek eldöntésére egy státusz regiszterben jelzem – többek között – egy bit billentésével, hogy átfordult-e már a végpointer (azaz a struktúra végéről megelőzte-e a kezdő mutatót) (25.ábra). Ennek tudatában eldönthető a probléma. Ha a kezdő mutató visszatér, törölöm a megfelelő bitet. Egyéb állapotokat is jelzek a státuszregiszterben, mint például az alulcsordulás, azaz ha szeretnék olvasni a bufferből, de nincs benne adat.

Miután egy egyértelműen működő struktúrában tárolhatók a szükséges adatok (megfelelően lehet írni és olvasni a memóriacímeit), további műveletek is adhatók az adatszerkezethez, melyek az állapotát kérdezik le, vagy az elemein végeznek a műveleteket. Esetemben két ilyen műveletet is megvalósítottam. A bufferben szereplő elemek átlagát kiszámító műveletet, és az aktuálisan a bufferben levő elemek számát lekérdező műveletet. Ezekre a funkciókra a korábban már említett számítások elvégzése miatt van szükség. A gyűrűs regiszter konkrét megvalósítását a mellékletben található rbf.asm nevű fájl tartalmazza.

Korlátvizsgálat

Ez a fejezet a referencia fog felismerésének konkrét módszerét mutatja be, amely talán az egész alkalmazás leglényegesebb eleme. A részfeladat célja az aktuális eltelt idő és a korábbi átlagok ismeretében megbecsülni azt az időintervallumot, amelyben vélhetően érkezni fog egy újabb fogaskerékfog által generált impulzus. Ehhez a már korábban meghatározott korlátok ismeretére és a fogaskerék paramétereire van szükség. A korlátozások között szerepelt a felismerhető maximális illetve minimális forgású fogaskerék körfrekvenciája, a fogaskerék átmérője, és a rajta levő fogak száma. Szükséges még a mozgás dinamikáját korlátozó feltevés, miszerint a maximális szögsebesség – egyenletes gyorsulást feltételezve – 3 teljes körfordulás alatt érhető el leggyorsabban, másrészt a minimális szögsebesség $\frac{1}{4}$ fordulat alatt. A feltételek felhasználásával meghatározható egy általános feltétel, mely a vélhető érkezés időintervallumát adja meg. A számítás a következő ábrán látható. **(26. ábra)**

A matekot részletesen magyarázni!

$$r = 100[\text{mm}] \quad x = 48[\text{darab}]$$

$$\omega_{\min} = 3[\text{Hz}] \Rightarrow \omega'_{\min} = \frac{3x}{360^\circ} = \frac{1}{16} \left[\frac{\text{fordulat}}{\text{s}} \right]$$

$$\omega_{\max} = 200[\text{Hz}] \Rightarrow \omega'_{\max} = \frac{200x}{360^\circ} = \frac{25}{6} \left[\frac{\text{fordulat}}{\text{s}} \right]$$

$$v_{\min} = 2r\pi\omega'_{\min} \approx 39 \left[\frac{\text{mm}}{\text{s}} \right]$$

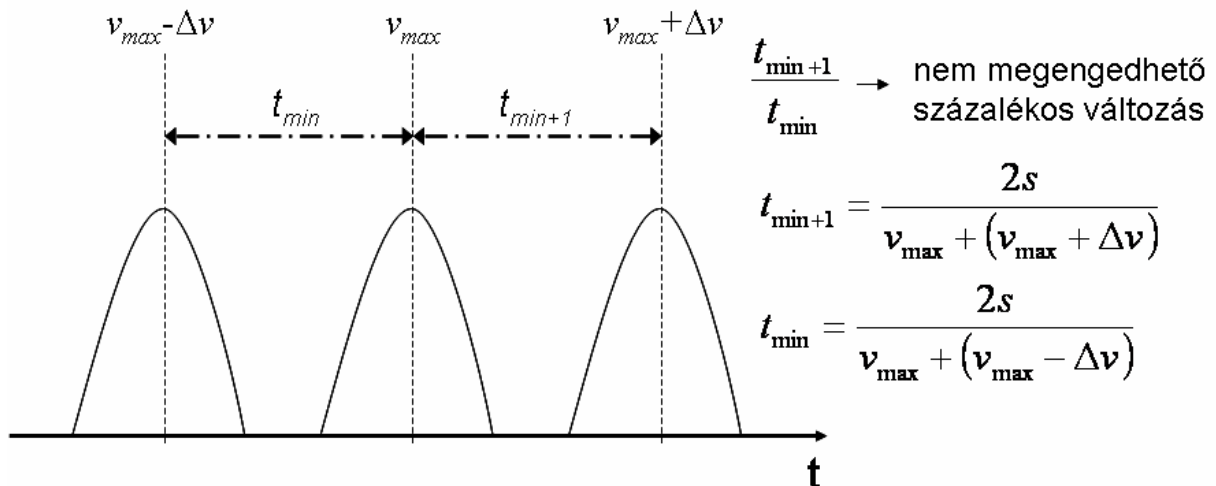
$$v_{\max} = 2r\pi\omega'_{\max} \approx 2617 \left[\frac{\text{mm}}{\text{s}} \right]$$

$$s = 3 \cdot 2r\pi \approx 1884[\text{mm}]$$

$$\Delta v = v_{\max} - v_{\min} = 2578 \left[\frac{\text{mm}}{\text{s}} \right]$$

$$\Delta v = 1,37 \left[\frac{\text{mm}}{\text{s}} \right]$$

$$\Delta v_{\text{fog}} = \Delta v \frac{2r\pi}{x} \approx 18 \left[\frac{\text{mm}}{\text{s}} \right]$$



$$\frac{t_{\min+1}}{t_{\min}} \approx 0,69\%$$

$$\frac{t_{\max+1}}{t_{\max}} \approx 41\%$$

26. ábra: A dinamika korlátozásáért felelős százalékos rendszerkonstansok számítása

A számítások eredményeként megkaptam a dinamikát leíró korlátokat. Az végeredmény azt mutatja, hogy bármilyen a feltételeknek eleget tevő mérések esetén leghamarabb az aktuálisan rögzített eredmény plusz 0,69%-ának megfelelő idő multával jöhet egy bizonyosan rákövetkező fog által generált új impulzus. A leghosszabb várakozási idő pedig 141%-nak megfelelő idő elteltéig tart. Ha nem ez történik, akkor két lehetőség áll fenn: az egyik a referencia fog elhaladása, a másik pedig, hogy valamely módon megszűnt a jelek továbbítása. Ez történhet akár a tekerés befejeztével is, illetve a készülék meghibásodása miatt is stb....

A korlátokat szintén igazítani kell a mikrokontroller által reprezentálható környezethez. Mivel sajnos az utasításkészletben nem szerepel osztás (nincs osztó egység), más lehetőségek közül kell választani. Én a közelítést választottam, mert talán a legkevesebb befektetéssel jár, és megfelelő eredményt produkál. Ez azt jelenti, hogy egy szám 0,69% kiszámítását 128-al történő osztással valósítom meg, mely az osztandónak pontosan 5-el történő jobbra shiftelését jelenti. Így eredményül az osztandó 0,8%-a adódik. Másrészről a szám 41%-ának kiszámítása egy jobbra shifteléssel történik, ami a szám 50%-át jelenti.

A megkapott konstansok felhasználásával számolhatók azok az intervallumok, amikben vélhetően megjelenik egy impulzus. Az eljárás a következő: az eltelt idők átlagának vesszük megfelelő százalékát, majd lassulás esetén az átlag plusz a százalék eltelte adja az időintervallum felső határát, gyorsítás esetén az átlag mínusz a százalék adja az intervallum alsó határát.

A feladat inentől egészen egyszerű, ellenőrizni kell, hogy az aktuálisan rögzített időadat milyen intervallumba esik. Ha a normál intervallumban van, akkor az egy rákövetkező fog. Ha kétszeresére bővítve az intervallumot esik csak bele az impulzus bekövetkezte, akkor még ellenőrizni kell, hogy az ezt követő impulzus a normál intervallumban van-e. Ha igen akkor a referencia fog haladt el, ha nem akkor a már említett esetek közül történhetett egy. Az itt közölt eljárás implementációja a BoundryCheck.asm nevű fájlban található.

Státusz regiszterek

A feladat megvalósításakor igyekeztem modulokra bontani az egyes részfeladatok alapján a kódot. Így kerültek külön fájlba az analóg/digitális átalakításért felelős blokk (ADconversion.asm), a soros kommunikáció beállításáért és végrehajtásáért felelős blokk (serialcomm.asm), a mikrokontroller konfigurációs bitjeit beállító, fordítási egységeket és az IT-ket meghívó blokk (main.asm), és a végeredményben a pedálszög számítását végző blokk (Angle.asm). Minden egyes fordítási egység külön deklarációval rendelkezik, különböző memória területeken foglalnak helyet a változók számára, azonban a kód egybefordítása nem takarja el ezeket egymástól. Mégis a biztonságosság kedvéért státusz regiszterek használatával történik a kommunikáció. A DYNAMIC_STATUS regiszter a korlátok vizsgálatának eredményéről tartalmaz információt, a BOUNDARY_FLAG regiszter az ellenőrzéskor ideiglenesen eredményeket tárol, a STATUSREG a ring buffer állapotait jelzi. A bitek definícióját a lefoglalásuk helyénél található megjegyzések tartalmazzák.

Illesztés a stimulátorral

A feladat által támasztott elvárások alapján a végső cél egy stimulátor vezérlése a mikrokontroller segítségével. A rendelkezésre álló MotionStim8 stimulátor elvi alapon alkalmas a mikrokontrollerrel történő vezérlésre, a Science Mode üzemmódban történő alkalmazás dokumentációja alapján.[16] A Single Pulse Mode megfelelő eszközöket biztosít az elektromos stimuláció elvégzésére.

Ennek megvalósításához a pedálszög információt a stimulátor számára érvényes és értelmezhető formára kell hozni, amely implicite magában foglalja a mérések során sikeresen meghatározott stimulációs módszerek implementálásának nélkülözhetetlenségét gépi szinten. A legmegfelelőbb erre a célra az OORI-ben is alkalmazott stimulációs mintázat.[7], mely alapján adott pedálszög esetén, a PIC kimenetén a stimulációs módszer által meghatározottan kell megjelennie egy a Single Pulse Mode-ra jellemző 4 bájtos bitsorozatnak. Az **27. ábra**, a vezérlő bájtok által kódolt információt mutatja, a **28. ábra** pedig a négy bájt pontos definícióját.

Variable	Bits	Value/Range	Description
Ident	2	3	Command identification number
Check	5	0..31	Checksum = sum of all logical variables Modulo 32 = (Channel_Number + Pulse_Width + Pulse_Current) modulo 32
Channel_Number	3	0..7	Channel_Number = 0 is channel no. 1 : Channel_Number = 7 is channel no. 8
Pulse_Width	9	0,10..500	Pulse width in μs
Pulse_Current	7	0..127	Current in mA

27. ábra: Az impulzusgeneráló utasítás változói

Byte	Bits	Value	Variable	Bit no. with respect to the variable
Byte 1	7	1		
	6	1	Ident	1
	5	1	Ident	1
	4		Check	4
	3		Check	3
	2		Check	2
	1		Check	1
	0		Check	0
Byte 2	7	0		
	6		Channel_Number	2
	5		Channel_Number	1
	4		Channel_Number	0
	3	X		
	2	X		
	1		Pulse_Width	8
	0		Pulse_Width	7
Byte 3	7	0		
	6		Pulse_Width	6
	5		Pulse_Width	5
	4		Pulse_Width	4
	3		Pulse_Width	3
	2		Pulse_Width	2
	1		Pulse_Width	1
	0		Pulse_Width	0
Byte 4	7	0		
	6		Pulse_Current	6
	5		Pulse_Current	5
	4		Pulse_Current	4
	3		Pulse_Current	3
	2		Pulse_Current	2
	1		Pulse_Current	1
	0		Pulse_Current	0

28. ábra: Az impulzusgeneráló utasítás bájtjainak definíciója

Amint látható, ennek a specifikációnak eleget tevő adatnak kell rendelkezésre állnia a kimeneten. Például ha a stimulátorral ki szeretnénk adni a 3-as csatornán (mivel a csatornák számozása eredetileg 1 – től 8 – ig megy és 3 bit áll rendelkezésre, binárisan 0-tól 7 –ig számozhatók rendre)200mikros-ig tartó 120 mA-es stimuláló jelet az a négy bájtban a következő képpen néz ki:

Byte 0 – 11100010 1 11 –SPM 00010 – CHK (mivel $2+200+120 \text{ mod } 32 = 2$)
 Byte 1 – 0010XX01 0 010 – CHA XX ...01 – PW
 Byte 2 – 01001000 0 1001000 – PW (11001000 ez előző bájtból
 $128+64+8=200$)
 Byte 3 – 01111000 0 1111000 – PC ($64+32+16+8=120$)

A stimulátor kommunikál a számára utasításokat küldő külső eszközzel, mely egy acknowledgement (nyugtázó) bájtban nyilvánul meg. Ennek is megvan a szintaktikája, amennyiben az utasítás végrehajtása során problémák adódnak ez a bájt 11000000, míg ha nincs semmilyen fennakadás a bájt értéke 1100001 (itt is a két egyes a sorozat elején a Single Pulse Mode-ban való működést jelzi).

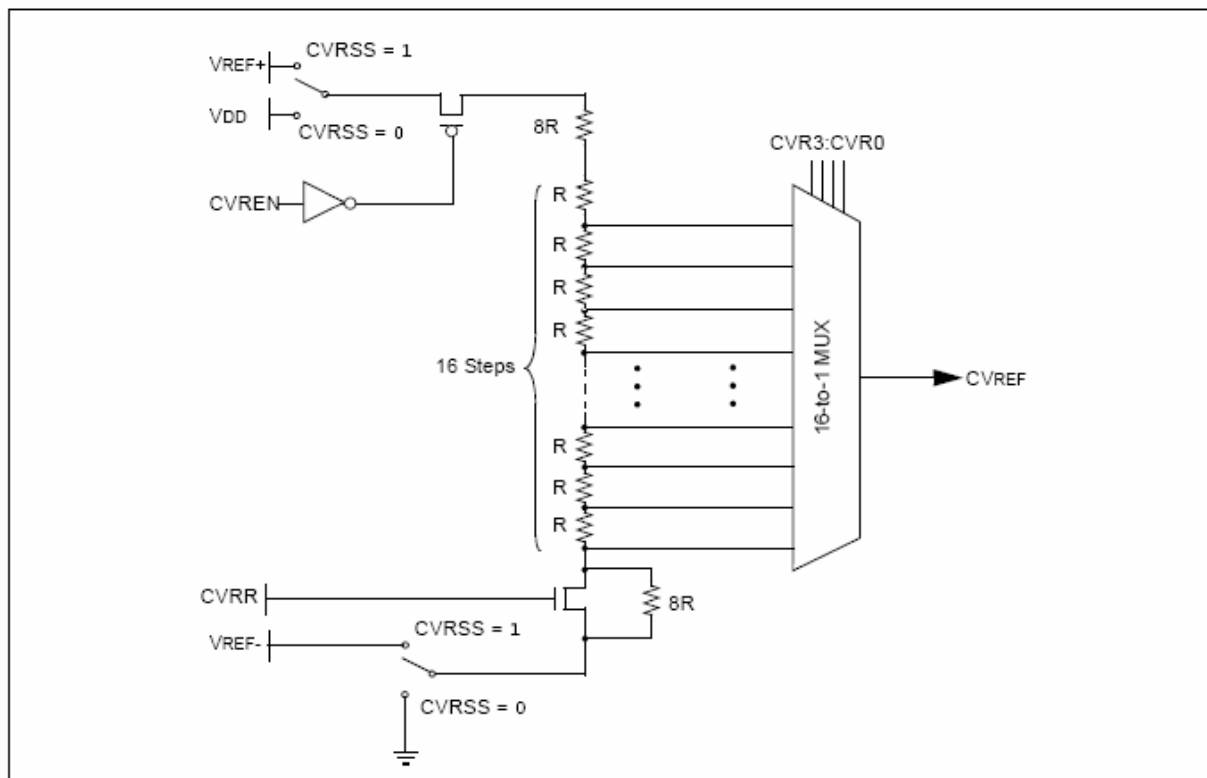
További fejlesztési lehetőségek

A diploma leadásának pillanatában a rendszer tesztelése a legfőbb feladat, mely aktuálisan is folyamatban van és minden bizonnyal a végéhez közeledik. Ezen felül több fejlesztési lehetőség is kínálkozik, melyek más-más irányokba terelhetik az alkalmazás fejlesztést.

Felmerült a kérdés, hogy mi történik, ha visszafelé tekerjük a pedált. Ekkor a szenzor jelében ugyanúgy keletkezik pozitív csúcs, és ez elhangolhatja az abszolút pozíciót. Tehát lehetséges ellenőrizni, hogy ha két azonos típusú (két pozitív vagy két negatív) csúcs következik egymás után az elvezetésben, akkor azt detektálva, felismerhető a tekerés irányának megváltozása. Ehhez minden szükséges feltétel adott, hiszen ez a feldolgozás (azaz a mikrokontroller) számára nem jelent speciálisabb elvárásokat. Amit meg kell tenni, hogy az előerősítő áramkört gyakorlatilag lemásolva „fejre kell állítani”. A dióda megfordításával ez elérhető.

Ha ez megvan, akkor a detektálás hasonló feldolgozást igényel, mint pozitív csúcsoknál. Izgalmas feladat ilyen módon a két feldolgozás összeillesztése.

Egy lehetséges megoldás, ha megvan a pozitív és negatív csúcsok detekciója, akkor valamilyen módon meg kell adni, hogy milyen amplitúdójú csúcsok érdekesek a számunkra. Ennek a megvalósítása sokféleképpen elképzelhető: adaptív úton meghatározni a határértéket, vagy átlagszámítás segítségével...stb. Arra azonban figyelni kell, hogy a zajba egyik módszerrel megállapított határértéknek sem szabad beleesnie. A mikrokontrollerrel lehetséges jó és változtatható referencia feszültség előállítása, ez volt az egyik jelentős paraméter, amit a kiválasztásánál figyelembe vettem. Tehát ehhez a referenciához hasonlítva lehet megállapítani, hogy az éppen aktuális bemeneten érkezett-e valamilyen csúcs. A referencia feszültség előállítására kétféle lehetőség adódik, amely az ábrán látható kapcsolás alapján könnyebben érthető (29. ábra).



29. ábra: Referencia feszültség előállításának blokkdiagrammja

Látható, hogy egyetlen bit (CVRR) beállításával választhatunk az üzemmódok közül. Ha a bit áll, akkor az alul látható ellenállás leválasztásával a tápfeszültség alsó kétharmadát tudjuk referenciaként felhasználni 1/16-os lépésközzel. Ez 5V-os

tápfeszültség esetén megközelítőleg 20mV-os komparálási felosztást jelent 0V-tól 3,3V-ig. Ha a bit nulla, akkor a tápfeszültség $\frac{1}{4}$ -től $\frac{3}{4}$ -ig terjedő tartományt osztja fel 16 részre, amely 5V-os tápfeszültség esetén 1,25V-tól 3,75-ig 0.15V lépésközzel teszi elérhetővé a referencia feszültséget. Ezek kiválasztása négy bit segítségével történik (CVR3:CVR0). A megfelelő, szenzoros jelhez igazított referenciát tehát ilyen pontossággal lehet kiválasztani. Ennek stratégiája és a további megfontolások nem tisztázottak.

Emellett, mint a legtöbb elektronikus alkalmazás ez is egy aktuális probléma megoldására törekszik, azonban a mai technikai elvárások megkövetelik a könnyű kezelhetőséget, az esztétikát, a robusztusságot és általánosságot. Ennek értelmében kijelző illesztése, a paraméterek külső állíthatóságának biztosítása egy elérhető cél, mely nem áll távol a beágyazott rendszerek világától. Ennek segítségével a rendszer adaptálható lenne bármilyen típusú kerékpárokhoz.

A gépi kód tovább egyszerűsíthető és optimalizálható az egyes műveletek utasításszámát figyelembe véve, illetve általánosabban leírva a műveleteket lecserélhetővé válna néhány kódismétlés. Ezen kívül pontosítani is lehetne az alkalmazást a közelítések helyettesítésével (pl. egy jól működő osztó algoritmus implementációjával).

Összefoglalás

Irodalomjegyzék

- [1] Kralj Alojz R., Functional Electrical Stimulation: Standing and Walking after Spinal Cord Injury, CRC Press, 1989
- [2] Taylor P., Esnouf J., Hobby J., „The functional impact of the Freehand System on tetraplegic hand function”, Clinical Results, Volume 40, Number 11, Page 560, November 2002
- [3] Taylor PN, BurrIDGE JH, Dunkerley AL, Wood DE, Norton JA, Singleton C, Swain ID. „Clinical use of the Odstock dropped foot stimulator: its effect on the speed and effort of walking” Arch Phys Med Rehabil., 80(12), 1577-83, 1999 Dec
- [4] Gerald E. Loeb, M.D.; Frances J. R. Richmond, Ph.D.; Lucinda L. Baker, P.T., Ph.D. The „BION Devices: Injectable Interfaces With Peripheral Nerves and Muscles”, Neurosurgical Focus, 20(5) 2006
- [5] GFÖHLER M.; LOICHT M., LUGNER P., „Exercise tricycle for paraplegics”, Medical & biological engineering & computing, , vol. 36, no1, pp. 118-121 (10 ref.) 1998
- [6] Kuohsiang C., Shih-Ching C., Kuen-Horng T., Jia-Jin J. C., Nan-Ying Y., Mao-Hsiung H., „An improved design of home cycling system via functional electrical stimulation for paraplegics”, International Journal of Industrial Ergonomics, 34 223–235, 2004
- [7] Pilissy T., „Lábmozgások kinematikai és bioelektromos jellemzőinek modellezése gerincsérültek és egészségesek kerékpározó mozgásánál”, diplomamunka, PPKE, 2007
- [8] Szécsi J, Fincziczki A, Laczko J, Klauber A „Functional electrical stimulation for cycling of paraplegics: individual adaptation of the clinical environment in Hungary”, 11th Annual Congress of the European College of Sport and Science, Book of abstracts, p.406.,2006
- [9] Szécsi J, Laczkó J, Klauber A, „Funkcionális elektrostimuláció segítségével hajtott tricikli és ergométer harántsérültek számára: a technikai környezet individuális illesztése a pácienshez”, Orvosi Rehabilitáció és Fizikális Medicina Magyarországi Társasága XXIV. Vándorgyűlése, Előadás összefoglalók, pp 48-50. 2005
- [10] Szécsi J, Fincziczki Á, Laczkó J, Straube A, „Elektrostimuláció segítségével meghajtott (háromkerekű) kerékpár: Neuroprotézis harántsérült páciensek mindennapos használatára” Rehabilitáció, 15. Évfolyam pp.9-14, 2005
- [11] Pilissy T, MSc; Klauber A, MD; Fazekas G, MD Laczkó J, MSc, PhD; Szécsi J, MD, MSc, „Improving functional electrical stimulation driven cycling by proper synchronization of the muscles”, Ideggyogy Sz, 61(5–6), 2008
- [12] T Pilissy, K Pad, G Fazekas, M Horvath, G Stefanik, J Laczkó, „The role of ankle-joint during cycling movement Task”, Conference article, CEFRR, 2007
- [13] Kovács F. Ferenc, A VLSI informatikai áramkörei, első kiadás, Budapest, Pázmány Egyetem Elektronikus Kiadó, 2004

Műszaki dokumentumok

[14] „MPLAB ICD 2 In-Circuit Debugger User's Guide”, Microchip Technology Incorporated DS51331A, USA, 2003

[15] „Microchip PIC18F2455/2550/4455/4550 Data Sheet” Microchip Technology Inc., DS39632D, 2007

[16]W. Scholz, K. Busch, „ScienceMode for the MOTIONSTIM8 - Description and Protocol”, 2004

Egyéb források

Az stimulátor honlapja: www.medel-hamburg.de

Az ergométer honlapja: www.motomed.de

Az 1. ábra képei a gyártók honlapjáról származnak

A 16. ábra [14]-ből, 18., 19., 20., 21., 22., 29. ábrák [15]-ből és a 27., 28. ábrák [16]-ből származnak

Ábrahivatkozások minden xx. ábra felíratnak legalább 2-szer szerepelni kell

Irodalom hivatkozásokat beírni a dokunemtumba.