



Budapesti Műszaki és Gazdaságtudományi Egyetem
Villamosmérnöki és Informatikai Kar
Méréstechnika és Információs Rendszerek Tanszék

Kalocsai Dávid

**SZÉN-DIOXID KIBOCSÁTÁS
MÉRÉSE ORVOSI
FELHASZNÁLÁSRA**

KONZULENS

Orosz György

BUDAPEST, 2012

Tartalomjegyzék

| | |
|--|-----------|
| Kivonat | 5 |
| Abstract | 6 |
| 1 Bevezetés | 7 |
| 2 Irodalomkutatás | 9 |
| 2.1 Elektronikus eszközök fejlődése az egészségügyben | 9 |
| 2.2 Légzés, szív és érrendszer | 9 |
| 2.2.1 Légzőszervi és szív- és érrendszeri megbetegedések | 9 |
| 2.2.2 Légzés mechanizmusa | 10 |
| 2.2.3 Spirometria | 11 |
| 2.2.4 Ergospirometria | 14 |
| 2.3 Szén-dioxid szenzor | 15 |
| 2.3.1 Szén-dioxid-mérési eljárások | 15 |
| 2.3.2 NDIR módszer részletes ismertetése | 18 |
| 2.3.3 NDIR szenzor kalibrálása | 25 |
| 2.3.4 Szén-dioxid szenzor további felhasználása | 27 |
| 3 Részletes specifikáció..... | 29 |
| 3.1 A rendszer részletes specifikációja | 29 |
| 4 Rendszerterv..... | 31 |
| 4.1 Logikai rendszerterv | 31 |
| 4.2 Hardverterv..... | 32 |
| 4.2.1 Mikrokontroller..... | 32 |
| 4.2.2 Külső A/D..... | 35 |
| 4.2.3 Szenzor és lámpa | 35 |
| 4.2.4 Kalibrációs szenzor | 37 |
| 4.2.5 Nyomásszenzor..... | 38 |
| 4.2.6 Pumpa..... | 39 |
| 5 Hardvertervezés | 40 |
| 5.1 Fejlesztői környezet | 40 |
| 5.2 Kapcsolási rajz..... | 40 |
| 5.2.1 Tápellátás..... | 40 |
| 5.2.2 Mikrokontroller..... | 41 |

| | | |
|----------|---|-----------|
| 5.2.3 | Szenzorok jeleinek erősítése..... | 42 |
| 5.2.4 | Külső A/D..... | 43 |
| 5.3 | Nyomatott áramkör elkészítése | 44 |
| 5.3.1 | Beültetési rajzok (footprintek)..... | 44 |
| 5.3.2 | Szabályok beállítása..... | 44 |
| 5.3.3 | Alkatrészek elhelyezése | 45 |
| 5.3.4 | Vezetékezés | 46 |
| 5.3.5 | Gyártófájlok elkészítése | 47 |
| 5.3.6 | Élesztés..... | 47 |
| 6 | Firmware | 49 |
| 6.1 | Firmware alapjai | 49 |
| 6.1.1 | Állapotok..... | 49 |
| 6.1.2 | Firmware egységei..... | 50 |
| 7 | PC-s szoftver | 55 |
| 7.1 | Szoftver részegységeinek bemutatása..... | 55 |
| 7.1.1 | Alap DLL-ek | 55 |
| 7.2 | Szoftver részegységeinek bemutatása..... | 56 |
| 7.2.1 | Soros porti kommunikáció | 56 |
| 7.2.2 | _Dataprocess és Dataprocess modul..... | 58 |
| 7.2.3 | _Properties modul | 58 |
| 7.2.4 | Calculation osztály..... | 59 |
| 7.2.5 | Charts | 59 |
| 7.2.6 | Szimulátor | 62 |
| 7.2.7 | Modulok összekapcsolása: DataCenter és Controller osztályok..... | 62 |
| 7.2.8 | Központi megjelenítő modul | 64 |
| 8 | Kalibráció és szenzor tesztelése..... | 67 |
| 8.1 | Külső kalibrációs széndioxid használata..... | 67 |
| 8.2 | Kalibráció környezeti levegő és kifújt levegő használatával | 71 |
| 9 | Összegzés, további fejlesztési lehetőségek..... | 73 |
| | Irodalomjegyzék..... | 74 |
| | Függelék..... | 76 |

HALLGATÓI NYILATKOZAT

Alulírott **Kalocsai Dávid**, szigorló hallgató kijelentem, hogy ezt a diplomatervet meg nem engedett segítség nélkül, saját magam készítettem, csak a megadott forrásokat (szakirodalom, eszközök stb.) használtam fel. Minden olyan részt, melyet szó szerint, vagy azonos értelemben, de átfogalmazva más forrásból átvettem, egyértelműen, a forrás megadásával megjelöltem.

Hozzájárulok, hogy a jelen munkám alapadatait (szerző(k), cím, angol és magyar nyelvű tartalmi kivonat, készítés éve, konzulens(ek) neve) a BME VIK nyilvánosan hozzáférhető elektronikus formában, a munka teljes szövegét pedig az egyetem belső hálózatán keresztül (vagy hitelesített felhasználók számára) közzétegye. Kijelentem, hogy a benyújtott munka és annak elektronikus verziója megegyezik. Dékáni engedéllyel titkosított diplomatervek esetén a dolgozat szövege csak 3 év eltelte után válik hozzáférhetővé.

Kelt: Budapest, 2012. 12. 14.

.....
Kalocsai Dávid

Kivonat

A diplomatervem célja egy olyan prototípus modul fejlesztése, amely képes szén-dioxid mérésére a kilélegzett levegőben.

A feladat megoldása végighalad egy hardver termék prototípusának fejlesztési lépésein: termék szükségesének indoklása és felhasználási területei elemzése, folytatva a specifikációs követelmények összegyűjtésével, a lehetséges megoldások elemzésével, hardver-szoftver szétválasztással, a kapcsolási rajz és a nyomtatott áramköri tervek elkészítésével, majd a panelélesztés és a hozzá tartozó mind beágyazott, mind számítógépes tesztelő szoftver előállításáig.

A műszerben az úgynevezett NDIR (Nondispersive Infrared Sensor) módszert alkalmazó TPS2534G2/G20 típusú szenzort használtam, amely a szén-dioxid infravörös sugarak elnyelési tulajdonságán alapul. Az alkalmazott Atmega324A mikrokontroller összegyűjti a környezeti paramétereket: megméri a nyomást a mintavételi csatornában és a szenzor hőmérsékletét, valamint begyűjti a szén-dioxid szenzor jeleit, és azokat a PC-nek továbbítja. A számítógépen futó szoftver az adatokat kielemez, megjeleníti és menti.

A dolgozat első felében összegyűjtöttem a szükséges ismereteket, melyek egyben meghatározzák az alapvető követelményeket, majd a második felében a prototípus megvalósítási lépéseit mutatom be egészen egy tesztelésére alkalmas PC szoftver elkészítéséig.

Abstract

The aim of this thesis is to develop a prototype module which is capable of measuring carbon dioxide concentration of exhaled air.

The thesis introduces the development of a prototype instrument step-by-step: justification of the product and analyzing it's field of use continued by collecting requirements of the system, comparing the possible solutions, separating the tasks into hardware and software part, performing the schematic and layout design for manufacturing, implementing firmware of the device and PC software, and finally testing of manufactured board.

The sensing of carbon dioxide concentration was performed using a sensor of type TPS2534G2/G20. This sensor is based on NDIR (Nondispersive Infrared Sensor) method which uses the property of carbon dioxide that it absorbs infrared light. The applied Atmage324A microcontroller performs the data acquisition: measures the pressure of the tube containing the exhaled air and the output signal of the carbon dioxide sensor, furthermore it sends these values to the PC. The software running on the computer analyzes, displays and saves the data received from the microcontroller.

The first part of the thesis gathers the necessary knowledge which also defines the basic requirements, and then the second part introduces the steps of realizing a hardware prototype as well as a PC software capable of testing the system.

1 Bevezetés

A dolgozatom célja egy NDIR (Nondispersive Infrared Sensor) módszeren alapuló szén-dioxid koncentráció mérésére alkalmas szenzor megépítése, valamint a hozzátartozó prototípus adatgyűjtő panel megtervezése és elkészítése.

Az okot az adta, hogy munkám során ergospirometriás (terheléses-élettani) készülékhez kellett keresnem oxigén és szén-dioxid érzékelőket. Ebből adódott az a tapasztalat, hogy a piacon hatalmas rés tátong az egyszerű szén-dioxidetektáló és a pontos koncentráció mérésére alkalmas eszközök között. Tovább csökkenti az orvosi felhasználásra alkalmas szenzorok számát, hogy a koncentrációmérésre alkalmasak közül sok lassan reagál az esetleges változásokra, mivel mérési csatornájukba a mérni kívánt összetétel diffúzióval áramlik be.

Az első féléves munka nagyobb részét az irodalomkutatás tette ki, mely során megvizsgáltam, hogy milyen egyéb orvosi méréseknél lenne használható egy ilyen eszköz, és azok milyen követelményeket támasztanának szenzorral szemben. Ezek után következhetett az elterjedt szén-dioxid mérésére alkalmas módszerek elemzése, összehasonlítása, melyek tudatában az NDIR módszer mellett döntöttem. Ez a módszer három komponensből áll: fényforrás, mintavételi csatorna/csatornák, valamint az érzékelő(k). Ezek kiválasztása hosszadalmas irodalomkutatás és szabadalmak átolvasása eredményeként jöttek létre. Ezek után a lehetséges kalibráció eljárásokkal ismerkedtem meg.

A fentiek együttes követelményrendszere egy teljes specifikációvá állt össze, mely mind a hardver, mind a szoftver valamennyi szükséges kritériumát meghatározta. Így a rendszerterv elkészítése után elkészült a szenzor vezérlésére alkalmas kapcsolási rajz, majd nyomtatott áramkör (NYÁK) terve. A kapcsolási rajznál törekedtem olyan alkatrészek választására, melyek a piacon könnyen beszerezhetőek. A nyákterv készítésénél különös figyelmet fordítottam az alapvető tervezési szabályok betartására, a mechanikai alkatrészek helyigényeinek követelményire, illetve az alkatrészek oly módon történő elhelyezésére, hogy azok könnyen, kézzel forraszthatóak legyenek.

Az elkészült tervek alapján legyártattam a nyomtatott áramkört, majd azt felélesztettem, és megírtam a firmware-t, mely alapvető funkciók elvégzésére képes:

megméri a szenzorok kimenetén lévő jeleket, és azokat USB-n keresztül továbbítja a PC felé, valamint vezérléssel kapcsolatos parancsok feldolgozására képes.

Legvégül elkészíttem a szoftvert, mely az USB-n kapott eredményeket feldolgozni, megjeleníteni és tárolni képes. Ezen kívül kalibrációs lépések elvégzésére is alkalmas.

A második fejezet az irodalomkutatás eredményeit összegzi, bevezeti, hogy miért is van szükség széndioxid szenzorra, milyen orvosi méréseknél használják. Majd összegyűjti a széndioxid szenzorokra vonatkozó fontosabb információkat: mérési módszereit és kalibrációjával is részletesebben foglalkozik.

Harmadik fejezetben a második fejezetben megjelent valamennyi követelményt összegzi, majd lefordítja azokat hardver és szoftver követelményre és így alakítja ki a specifikációt.

Negyedik fejezetben a specifikáció alapján elkészül a rendszerterv, melyből megtörténik a hardver-szoftver szétválasztás és kialakul a hardverterv. Ezt követi a fontosabb hardver egységek bemutatása: széndioxid szenzor lámpája, érzékelője, valamint a nyomásszenzor, a külső A/D, kalibrációs széndioxid és a mikrokontroller.

Ötödik fejezet bemutatja a hardvertervezéshez felhasznált fejlesztő környezetet és annak használatával bemutatja a hardvertervezés egyes fázisait, lépéseit, valamint a választott megoldásokat és annak indoklásait.

Hatodik fejezet a mikrokontrollerre fejlesztett firmware-t mutatja be. Felvázolja annak működési ábráját, majd egyes részegységek feladatát tárgyalja. A hardver és szoftver között közlekedő csomagok funkciói és felépítése is ebben a fejezetben kapott helyet.

Hetedik fejezet a PC-s szoftvert elemzi ki. Fejlesztésének lényege, hogy az előzetes követelmények megléte után kezd csak a tényleges modulok kifejlesztését, melyek egymástól teljesen függetlenül képesek teljes funkciók ellátására és csak a végső stádiumban lettek összekapcsolva.

Nyolcadik fejezetben található az épített szenzor kalibrációjára próbálkozások és példák, amely egyben annak ellenőrzése is, hogy képes-e széndioxid mérésre.

Kilencedik fejezetben található az értékelés, összefoglalás és a lehetséges következő lépések bemutatása.

2 Irodalomkutatás

2.1 Elektronikus eszközök fejlődése az egészségügyben

Az OECD (Gazdasági Együttműködési és Fejlesztési Szervezet) adatai alapján a tagok egészségügyi kiadásai a GDP arányában a 2000. évi átlagosan 7,8%-ról 9%-ra nőttek 2008-ra. A magas számok és az utóbbi években tapasztalható gazdasági helyzet a költségek drasztikus csökkentésére ösztönözi az országokat, melynek a növekvő társadalmak és lakosság előregedése komoly gátat szab. Ennek köszönhetően hatalmas piac nyílt az olyan eszközök számára, melyek hosszútávon a kiadások optimalizálásához vezethetnek: homecare (otthoni felügyeleti) termékek, melyek a rendelési óraszámokat csökkenthetik, valamint a komplex szűrő berendezések elterjesztése. Ezekre jó példa a mágnesesrezonancia-képpalkotás (MRI – Magnetic Resonance Imaging) berendezések, melynek száma 2000 óta több mint duplájára nőtt, valamint az Amerikában fejlesztett mobiltelefonokra szerelhető otoszkóp (fültükör) és dermatoszkópok (bőrre helyezhető mikroszkóp) [1][2].

2.2 Légzés, szív és érrendszer

2.2.1 Légzőszervi és szív- és érrendszeri megbetegedések

A földfelszín közelében a levegő összetétele: 78% nitrogén, 21% oxigén, valamint 1% nemesgázok, szén-dioxid és vízgőz. Bármilyen olyan anyag, ami ezen kívül megjelenik a levegőben, szennyező anyagnak minősül, tehát károsan hat a szervezetünkre. A légszennyezés miatt légzésszervi betegségek alakulnak ki, többet között asztma, COPD és a tüdőrák [3].

Ismertebb légzőszervi betegségek:

- *Asztma*: Asztmás beteg légútjai nagyon érzékenyek, akár hétköznapi ingerekre is a légutak összehúzódással reagálhatnak, emiatt a beteg nem tud friss, oxigén dús levegőt beszívni. Magyarországon az ösznépeség 3%-ra tehető számuk, de számuk folyamatosan nő, valamint a kezeletlen légúti allergiások 20-40%-nál alakulhat ki idővel [4].

- *COPD*: A krónikus obstruktív légúti betegség (*COPD*) az egyik fő halálozási ok a világon. WHO (World Health Organization) 2004-ben 64 millióra becsülte a betegségben szenvedők számát és 3 millióra az adott évben a betegség következtében elhunytakét világszerte. A COPD miatt visszafordíthatatlanul károsodnak az alsó légutak. WHO becslése szerint 2030-raminden harmadik ember haláláért lesz felelős [5].
- Tüdőrák a tüdőben növekvő rosszindulatú daganat. Világszerte a legtöbb elhalálozásért felelős daganatos betegek esetében. Az egyik leginkább megelőzhető daganatféleség, mivel 85-87% az aktív, míg 3-5% passzív dohányosoknál alakul ki. Kezelése történhet műtéttel, radioterápiával, kemoterápiával és célzott terápiával [6].

Szív és érrendszeri megbetegedések a világban évente 17,3 millió ember haláláért felelősek. Ez a szám WHO (World Health Organization) szerint 2030-ra elérheti a 23,6 milliót. A mutatók Magyarországra nézve sem kedvezőek az elhalálozások 54%-ért felelősek. A szívet és a koszorúereket megtámadó betegségek közül kiemelkedik a magas vérnyomás, a szívkoszorúér-megbetegedése és az infarktus [7].

Az érrelmeszesedés a kis és nagy vérerek öregedésével együtt járó folyamat, amely következtében a vérerek megvastagszanak és megmerevednek. Rugalmasság csökkenéséhez az érfalakban lerakódó koleszterin nagyban hozzájárul, amelyhez véralvadék is hozzátapad, így vérrög alakulhat ki, mely részben vagy teljesen elzárja az artériát, megakadályozva a vér áramlását.

A magas vérnyomás jelentős megterhelés ró a szívre, melynek következtében a szívelégtelenség alakulhat ki, mely abban nyilvánul meg, hogy egyre kisebb megterhelések esetén sem tud a szív megfelelő vért pumpálni az izom számára, a szervezet oxigénellátása zavart szenved [8].

2.2.2 Légzés mechanizmusa

Az emberi testnek folyamatos friss oxigéndús levegőre van szüksége, melyet a környezetünkből tudunk felvenni. A hozzá tartozó gázcserét a légző rendszerünk hatja végre, melynek főbb részei a felső és alsó légutak, tüdők, rekeszizom, légzési

segédizmok, légzőközpont alkotják. A tüdőnek nincs saját izomzata, ezért a mellkas és a rekeszizom mozgása hozza létre a légcserét. Légzés esetében kétféle légzést különböztünk meg: külső légzés, mely a környezet és a szervezet, míg a belső légzés a vér és a sejtek között történik.

Légzés során a levegő összetétel megváltozik. A levegő 21% oxigént, 0,04% széndioxidot, 78% nitrogént, és néhány tized százalék nemes gázt tartalmaz, míg a szervezetből kiáramló összetétele 16% oxigént, 4,04% széndioxidot tartalmaz. A nitrogén és a nemesgázok százalékos aránya érdemben nem változik. A többlet széndioxid az oxidáció élettani folyamata során a sejtekben keletkezik.

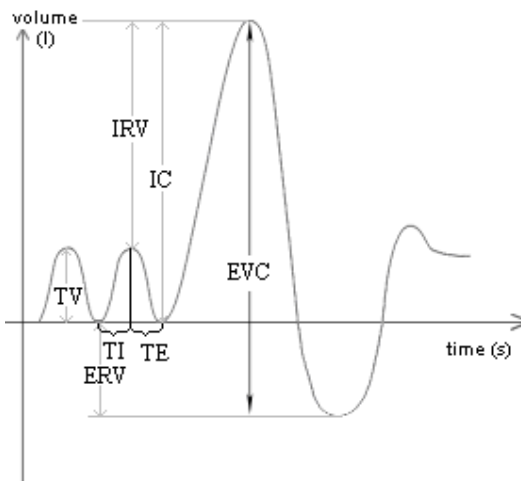
Légzés esetén kétféle szabályozásról beszélhetünk: automatikus (akaratától független) és akaratlagos szabályozásról. Az első esetben a szabályozást az agytörzs végzi, melynek fő kémiai ingere a széndioxid: amikor a szervezetben felhalmozódik, fokozódik a légzés, míg ha kiürül, csökken a légzés. Az akaratlagos szabályozás az agykéreghez kötött. A szabályozásban részt vesz a vegetatív idegrendszer is, mely fokozott igénybevétel esetén fokozza a légzés mélységét és frekvenciáját, míg alvásnál csökkenti azokat [9][10].

2.2.3 Spirometria

Egy hathetes csecsemő átlagosan 30 – 60 levegővételt vesz percenként. Ez az érték a korral együtt csökken, egészen a 12 – 14 éves korig, amikor átlagosan 12 – 16-szor lélegzünk percenként ki-be, mely során 500 ml levegő áramlik ki-be a tüdőnkől. Ebből következik, hogy normál felnőttél a légzési pertérfogat (VE) 8 liter körül van. Ez azért érdekes, mert a tüdő maximális kapacitásának egészséges értéke 4–5 l. Ezek és hasonlóan fontos paraméterek mérésére szolgálnak a spirometriai vizsgálatok, amelyek a legalapvetőbb műszeres vizsgálatok a légzésfunkció állapotának pontos megállapítására. Ezen értékek segítségével következtetni lehet a légzésben résztvevő tüdőrészek térfogatára, illetve a légutak állapotára, esetleges szűkületére. Fontos kiemelni, hogy a vizsgálattal már a kezdeti stádiumban is felismerhető egy esetleges légzésfunkciós betegség, valamint ellenőrizhető előrehaladottsága vagy a kezelés hatékonysága.

A nyugalmi vizsgálatot a vitálkapacitás (VC) mérése jelenti. A vitálkapacitás nem más, mint két-három egymást követő nyugalmi légzés után egy maximális

belégzést követő maximális kilégzés mértéke (2.1. ábra). Fontos, hogy a be- és a kilégzést lassan végezzük. A belégzési tartalék a maximális belégzéskor még felvehető levegő, amelynek mértéke 2000-2500 ml. A maximális kilégzéskor még kilélegezhető levegő a kilégzési tartalék, amelynek mértéke 1000-1500 ml körül mozog. E három levegőfrakció együttesen adja a vitálkapacitást, amely felnőtt embereknél mintegy 3500-4500 ml-t jelent. Mértéke természetesen függ a nemtől, a kortól, a testfelülettől.

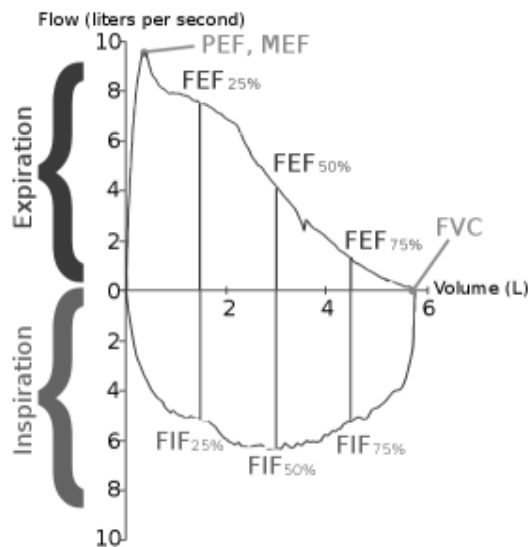


2.1. ábra: VC görbe. Forrás: [11]

VC görbe paramétereit:

- VC: Vitál kapacitás
- IVC: Inspirációs vitálkapacitás
- EVC: Expirációs vitálkapacitás
- IC: Inspiratórikus kapacitás
- ERV: Exspiratórikus rezerv
- IRV: Inspiratórikus rezerv
- TV: Légvétel térfogat
- TI: Inspiráció alatt eltelt idő
- TE: Expiráció alatt eltelt idő

Másik szélesebb körben használt mérési eljárás az erőltetett vitálkapacitás. Ez is néhány nyugalmi léggzéssel kezdődik, majd egy mély belélegzés után a kilélegzésnél törekedni kell a minél gyorsabb kilégzésre (2.2 ábra).



2.2. ábra: FVC görbe. Forrás: [11]

FVC paramtérei:

- FVC: Erőltetett vitálkapacitás
- PEF: Erőltetett kilégzési csúcsáramlás
- PIF: Erőltetett belégzési csúcsáramlasi
- FEV1: Erőltetett kilégzési másodperctérfogat
- FEV6: Erőltetett kilégzési térfogat 6 másodpercben
- FET: Forszírozott kilégzési idő
- FEF25%: FVC erőltetett belégzés 25%-nál mért áramlás
- FEF50%: FVC erőltetett belégzés 50%-nál mért áramlás
- FEF75%: FVC erőltetett belégzés 75%-nál mért áramlás
- FIF25%: FVC erőltetett belégzés 25%-nál mért áramlás
- FIF50%: FVC erőltetett belégzés 50%-nál mért áramlás
- FIF75%: FVC erőltetett belégzés 75%-nál mért áramlás

- MMEF25%-50%: Maximális középkilégzési áramlás 25% és 50% között
- MMEF25%-75%: Maximális középkilégzési áramlás 25% és 75% között

Ne felejtjük azonban, hogy a szervezet számára a ventiláció csupán eszköz, a cél a sejtek oxigénellátásának biztosítása. Az oxigénfelvételt viszont élettani körülmények között nem a légzés, hanem a keringés határozza meg, és a szervezet munkavégző képességét jellemzi. Az ideális állapot az, ha a rendelkezésre álló légzési térfogattól minél több oxigént tud a szervezet felvenni [11].

2.2.4 Ergospirometria

Az ergospirometria egy olyan diagnosztikai procedúra, mely során terhelés közben vizsgálják a páciens spirometriai és anyagcsere (gáz) folyamatait. Ezek alapján pontos képet kapnak a szív-, keringési- és a légzőrendszer, valamint az anyagcsere-működés állapotáról. A vizsgálat közvetve alkalmas az aktuális egészség és a teljesítőképesség felmérésére, élettani mutatók vizsgálatára és értékelésére. A különböző sportági mozgások, illetve munkakörök sajátos terhelésprofilal bírnak, amelynek figyelembevétele a célirányos vizsgálatok során elengedhetetlen.

A terhelést ergométerek segítségével érik el, mely lehet kerékpár, futószalag, vagy evezőpad. Európában a kerékpár, míg az Egyesült Államok területén a futószalagot használják leggyakrabban, ezek mérési eredményei nem hasonlíthatóak össze egy az egyben. Kutatások szerint a kerékpár pontosabb képet ad, mivel kevésbé számítanak olyan testi adottságok, mint például a páciens súlya. Az ergométer legfontosabb tulajdonsága, hogy milyen protokollokat ismer, amely nem más, mint hogy milyen módon változtatja a terhelés. Kétféle ismert: Az első esetben a terhelés konstans, ezt főleg lábadozó, transzplantáción átesett beteg esetén használják a gyógyulási állapot meghatározása, míg a második esetben alacsony szintről indulva folyamatosan növelik a terhelés értékét, ekkor a cél maximális teljesítőképesség meghatározása.

Mérés során mért legfontosabb paraméterek:

- VO_2 -oxigénfelvétel [l]
- VCO_2 -széndioxid kibocsátás [l]

- HR - pulzus [1/min]
- Spirometriai értékek
 - VE - Egy perc alatt kifújtt levegő [l/min]
 - Vt - Ki-belégtett levegő mennyisége [l/min]
 - RR - légtési frekvencia [1/min]
- WR - terhelés [W]
- RER - Oxigénfelvételt/Szén-dioxid kibocsátás

A teljes kifáradásig tartó terhelés során mért egyik legfontosabb paraméter, amely az aerob állóképességnek lényeges indikátora, a *maximális oxigénfelvétel*. Minél nagyobb a maximális oxigénfelvétel képesség, annál jobb az aerob állóképesség. Limitáló faktorai közé tartozik a légzőrendszer és a keringés oxigénfelvétel és szállító kapacitása, valamint az izmok oxigénfelvétel képessége is.

A szervezet energiaforgalmára jellemző adat az úgynevezett respirációs kvóciens (RQ), amely a belső légtés során a szövetekben termelődő szén-dioxid és a felhasznált oxigén arányát mutatja. Jelentősége abban áll, hogy a terhelés minden percében nyomon követhetők az anyagcsere-folyamatok. A nyugalmi-terhelés előtti- RQ akkor megfelelő, ha értéke 0,8 körüli értéket mutat. Terhelés kezdetén értéke csökken, mely arra utal, hogy az energiát a zsírokból ($R=0,71$), majd lassan növekedni kezd, ekkor azt szénhidrátból nyeri ($R\sim 1$) a szervezet.

2.3 Szén-dioxid szenzor

A szenzorok feladata valamilyen mennyiség (mechanikus, kémiai, termikus, mágneses, optikai stb.) villamos jellé történő átalakítása. Ezen jel felhasználás során legtöbbször erősítés és szűrés után egy feldolgozóhoz jut. Szenzor főbb osztályozása: aktív vagy passzív, lineáris vagy nemlineáris kimenetű, digitális vagy analóg.

2.3.1 Szén-dioxid-mérési eljárások

Széndioxid detektálására alkalmas eljárások két nagy csoportjával ismerkedtem meg:

- Kémiai elven működő CO₂ szenzorok,
- Optikai elven működő CO₂ szenzorok,
 - Akusztikus elven működő CO₂ szenzorok,
 - Raman szórás módszere,
 - Tömeg spektroszkópia,
 - Infravörös spektroszkópia: NDIR

2.3.1.1 Kémiai elven működő CO₂ szenzor

Ezen elven működő érzékelők egy speciális érzékelő réteggel vannak ellátva, amelynek alapja polisziloxin. Nagy előnyük a kis méretük valamint nagyon alacsony energiafogyasztásuk. Hátrányuk: a polisziloxin réteg gyorsan öregedik, így az érzékenységük, kimeneti jelük csökken, ami hosszútávon jelentős driftet eredményez. Fontos még, hogy a kémiai szenzorok „kikapcsolása” nehezen megoldható, csak légmentesen elzárt térben valósítható meg. Hasonló elven működő O₂ érzékelők 5-6 havonta cserélendők [12].

2.3.1.2 Optikai elven működő CO₂ szenzorok

Valamennyi optikai elven működő szenzor alapja, hogy kölcsönhatás alakul ki a fény elektromágneses tere és az általa megvilágított anyagban lévő töltések között. Ezen eljáráson alapuló szenzorok három fő részből állnak: egy optikai forrásból, valamilyen hullámhossz-beállítóból vagy szűrőből, valamint egy detektorból.

Bizonyos gázok, mint például a CO₂ és a N₂O képesek infravörös fényt elnyelésére. Elnyelődés hatására megnő a molekula mérete, amely következtében a gázban a nyomás megváltozik. Ezt használják ki a *fotoakusztikus* mérés során: ha az infravörös forrást impulzusszerűen kapcsolgatjuk, akkor az infravörös sugárzásból is impulzusszerűen vesz fel energiát a gáz, így a nyomás is ingadozni fog. Ezt a nyomásingadozást detektálják mikrofonnal és becsülik meg a csatornában lévő gáz CO₂ tartalmát. Nagy előnye, hogy nullpont kalibrációra nincs szükségünk, mert a nulla szint a „nincs infravörös fényt elnyelő anyag a csatornában” értéknek felel meg [13].

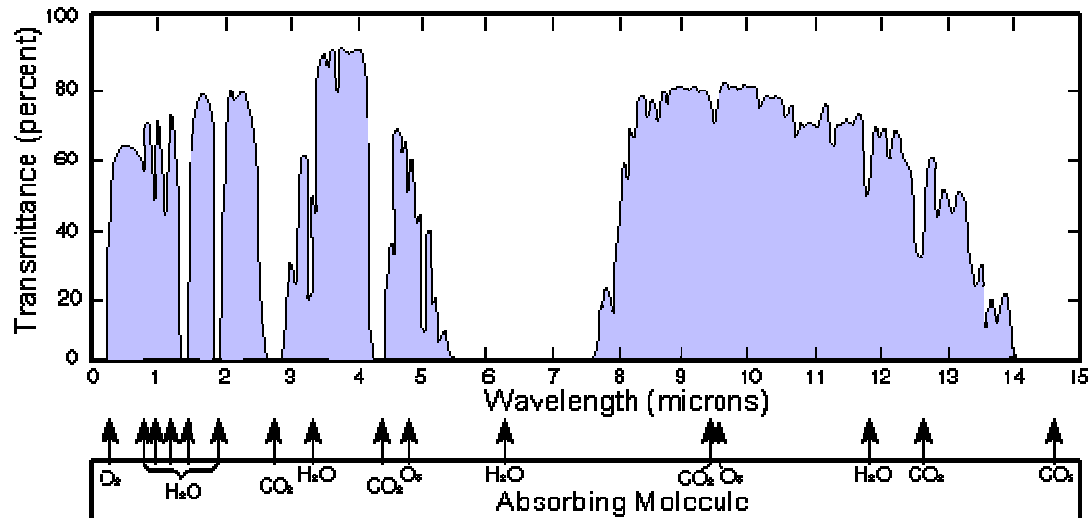
Raman-szórás esetén a fényforrást magas intenzitású monokróm argon lézersugár biztosítja. A lézer egy molekulához érve rugalmas vagy rugalmatlan

kölcsönhatásba lép a molekulával, mely során a fény visszaverődik, áthalad, elnyelődik vagy szétszóródik. Raman-szórásnak a foton molekulán történő rugalmatlan szóródást nevezzük, mely során a foton energiát ad át molekulának vagy vesz fel onnan. Ennek hatására a molekula forgási, rezgési és elektrongerjesztési energiája egyaránt megváltozhat. Eredendően ez egy gyenge folyamat, de a lézerek elegendő teljesítményt biztosítanak a spektrum rutinszerű méréséhez. A Raman-szórás spektrumvonalai segítségével minden molekula típusát pontosan azonosítani tudjuk [14].

Tömeg spektroszkóp működésének alapja, hogy a molekulákat szétválogatja tömeg/töltés hányados alapján. A gázt egy vákuumkamrába áramoltatjuk, ahol elektronsugárral ionizáljuk, és a mintát alkotóira bontjuk. Az ionokat mágneses mező gyorsítja fel a végső kamra felé, aminek mágneses mezeje merőleges a gáz gőzének útjára. Ennek hatására a különböző tömeg/töltés hányadosú komponensek, különböző görbületen fognak továbbhaladni. Ezután a komponenseket egy detektor lemezre ütköztetjük, ami érzékeli a komponenst, valamint annak koncentrációját is [15][16].

Mint már szó volt róla vannak molekulák melyek képesek elnyelni a fény bizonyos hullámhosszú komponenseit. Ilyen a CO_2 is, ami a $4.26 \mu\text{m}$ hullámhosszú infravörös sugarakat nyeli el. Nézzük meg az alábbi felépítést: vegyünk egy csatornát, amiben különböző gázokat áramoltatunk, a csatorna két oldalára pedig elhelyezünk egy infravörös fényforrást, míg a csatorna másik végére egy $4.26 \mu\text{m}$ szűrővel ellátott érzékelőt helyezünk, így az érzékelő csak az adott hullámhosszú fénysugarak érzékelésre alkalmas. Az így kapott eszköz tehát a csatornában lévő CO_2 tartalom mérését teszi lehetővé, a mért érték annál kisebb, minél nagyobb a koncentráció a csatornában. Ezt a *NDIR módszernek* nevezzük.

A bemutatott eljárások közül az NDIR az egyik legegyszerűbb, így az ára is kedvezőbb, valamint feladat bonyolultságának ez felel meg a legjobban. Fontos kiemelni, hogy ez egy általános eljárás. Tehát ha a CO_2 koncentrációjának mérésére alkalmassá tesszük az eszközt, akkor minimális munkával, a sávszűrő cserélgetésével, egyéb gázok koncentrációja is megmérhető. NDIR elven detektálható elemek és az általuk elnyelt IR hullámhosszak láthatóak a 2.3 ábrán.



2.3. ábra: NDIR elven mérhető gázok. Forrás: [17]

2.3.2 NDIR módszer részletes ismertetése

Az NDIR (Nondispersive infrared) szén-dioxidérzékelő működéséhez három komponensre van szükség: fényforrásra, mintavételi csatornára, és érzékelőre. A fény áthalad a mintavételi csatornában lévő levegőn és eléri az érzékelőt, amely a rajta lévő szűrőnek köszönhetően csak a megfelelő hullámhosszal rendelkező komponenst engedi át. Mivel a széndioxid elnyeli 4,26 μm hullámhosszú komponenseket, ezért az érzékelőig eljutó fény függ a csatornában lévő koncentrációtól, annak növekedésével csökken az értéke. Tehát egy széndioxidmentes és egy széndioxidot tartalmazó csatornában elvégzett mérés után a két állapot között lévő koncentráció különbség arányos az érzékelőn mért két érték hányadosával. A pontos számítást a Beer-Lambert egyenlőség fejezi ki:

$$I = I_0 * e^{-kP}$$

I : a mért intenzitás

I_0 : a mért intenzitás üres csatorna esetén

k : csatornára jellemző szám

P : a keresett koncentráció

Az egyenlőségből egyből kitűnik, hogy a keresett P mellett, ismeretlen a k érték is. Ez viszont az adott csatornára jellemző szám, tehát a kalibrációs eljárás legfontosabb feladata a k értékének meghatározása, melyről fentiek alapján tudható, hogy egy negatív szám lesz. Ezek után az alábbi egyenlőség adódik a koncentráció meghatározására:

$$P = \frac{1}{k} \ln \frac{I}{I_0}$$

2.3.2.1 Fényforrás

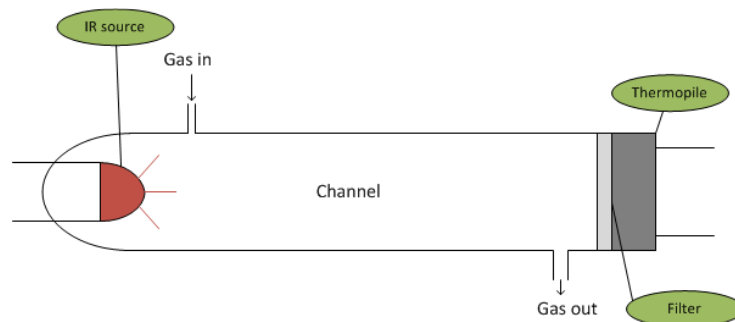
Fontos a lámpáról megjegyezni, hogy széles tartományban bocsát ki fényt, egészen a látható tartománytól az IR felső határáig. Ez nem okoz problémát, az érzékelőn lévő szűrő megfelelően működik, de a fényforrás által leadott enyhén változó hő figyelembe kell venni a mérési eredmény kiértékelésekor.

NDIR szenzorok egyik hátránya, hogy offszettel rendelkeznek, mely a használat során szépen lassan növekszik. Ez főként a lámpa öregedéséből származik, egyre kisebb energia kisugárzására képes, valamint az esetleges koszolódás is szerepet játszik ebben.

2.3.2.2 Csatorna

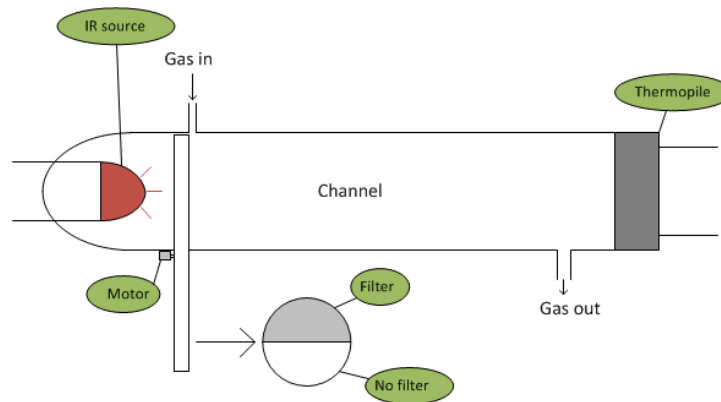
Csatorna ismertebb kialakításai:

- *Egyszerű egycsatornás*: Ez a legegyszerűbb, legkényesebb megoldás. A forrás és a nyelő a csatorna két ellentétes végén helyezkedik el. Ez egyszerűen megépíthető és legolcsóbb a kialakításának költsége is, de hátránya, hogy I_0 megméréséhez ki kell üríteni a csatornát, majd a mérés során ingadozó nyomás valamint hőmérséklet is hibát fog okozni a mérésben.



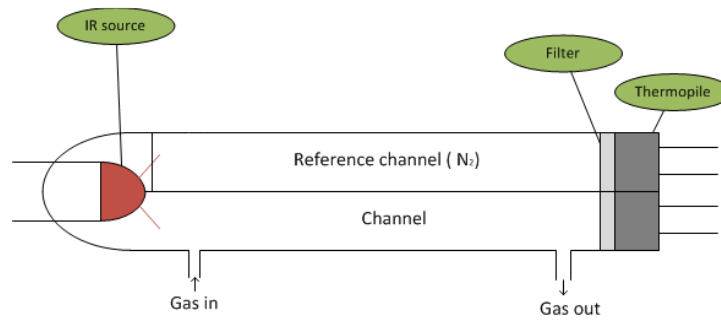
2.4. ábra: Egyszerű egycsatornás kialakítás

- *Egycsatornás változtatható szűrővel:* Ebben az esetben a forrás vagy a nyelő előtt egy kör alakú lemezt helyezünk el. Ennek egy szelete lóg be az adott elem elé. A kör területén egy optikai szűrő van felhelyezve, mely a $4,26\ \mu\text{m}$ -es szűrő, és egy lyuk is található rajta, amely teljesen átengedi a sugarakat. A kör lemezt forgatva kapjuk az I és I_0 eredményeket. Itt viszont komoly jelfeldolgozásra van szükség, és a forgás miatt zajosabb jelünk lesz, ami miatt több mérés átlagát érdemes venni.



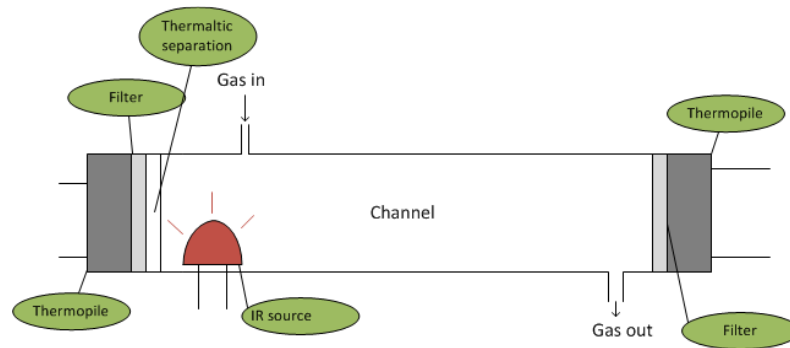
2.5. ábra: Egycsatornás változtatható szűrővel

- *Kétszatornás, referencia csatornával:* forrás és két nyelő között két darab csatorna helyezkedik el egymás mellett. Az egyik nyelő a mintavevő csatornánkat méri, amibe a CO_2 -t áramoltatjuk, míg a másik nyelőben egy referencia gáz van, amely nem tartalmazza CO_2 -t. Ez a referencia csatorna folyamatosan szolgáltatja számunkra az I_0 -t, amíg a másik csatornában mérjük az I -t. Hátránya a referenciagáz tökéletes lezárása valamint a két csatorna szimmetrikus kialakítása nem egyszerű mechanikai feladat, valamint a két azonos nyelő különböző hőmérsékletű lesz és gyártási pontatlanságaikból adódó hibákat is kompenzálni kell a mérés során.



2.6. ábra: Kétsatornás, referencia csatornával

- *Egycsatornás, két érzékelős:* Egyszerű egycsatornástól abban különbözik, hogy a forrás mellé is kerül egy nyelő, amelynek feladata az I_0 folyamatos meghatározása. Mivel ez elzártan helyezkedik el a fényforrás mellett így nulla széndioxid szintnek megfelelő értéket mér majd. Viszont a lámpa mellett van, ami folyamatosan melegíti. A két nyelő hőmérsékletkülönbsége kiemelkedő lesz [18].



2.7. ábra: Egycsatornás, két érzékelős

- *Egy csatorna két érzékelő, egy tokban:* Az elmúlt pár hónapban jelent meg egy új termék, mely egyetlen tokban tartalmaz két nyelőt, amelyből az egyik az I_0 , míg a másik az I mérésre alkalmas. Mivel egy tokban vannak így a hőmérsékletük közel azonos lesz. Szimmetrikus felépítésből adódóan a lámpa helyes elhelyezése esetén pontos mérési eredmények szülehetnek.



2.8. ábra: Egy csatorna két érzékelő egy tokban

A csatorna anyagának megfelelő megválasztásakor fontos annak fényvisszaverő tulajdonsága, mely az érzékelőhöz eljutó fény mennyiségén ötszörös erősítést is eredményezhet. Drágább eszközök belsejét arannyal, általános felhasználás esetén egyszerű alumínium vagy réz ajánlott. A csatornában áramló levegő be és kivezetését a két ellentétes oldalra a bevezetést a fényforráshoz, míg a kivezetést az érzékelő elé érdemes tervezni, ezzel biztosítva, hogy az áramló közeg biztosan áthaladjon a fényforrás és az érzékelő előtt. A csatorna hosszát 5 centiméter körülire érdemes tervezni.

2.3.2.3 Érzékelő–Thermopile

Az érzékelő működésének az úgynevezett *thermoelektromos-effektus* vagy más néven fordított Seebeck-hatás az alapja: Ha két különböző tulajdonságú fémeket egyik végükön egymáshoz kapcsolunk (hőelem), akkor a hőmérsékletkülönbségüktől függő feszültség alakul ki rajtuk. A hőelemen mérhető feszültség szint nagyon alacsony, de többet sorba kapcsolva az értékek összeadódnak, és így megfelelő szint érhető el. Az így kialakított eszközt thermopile-nak nevezzük.

Ahhoz, hogy az infravörös sugarakat is érzékelni tudjuk a „hot” oldalát a hőelemeknek infravörös elnyelő réteggel vonják be, így megvilágítás során a „hot” oldal felmelegszik és a kialakuló feszültség az besugárzott intenzitással lesz arányos.

Thermopile-ok legfontosabb tulajdonságaik:

- Állandó intenzitású sugárzást is érzékelik, nem kell a forrást modulálni, mint a piroelektromos detektoroknál. Ez nagyban megkönnyíti a mért jel kezelését.

- *Válaszkészség*: A válaszkészséget V/W-ban adják meg és mindig az érzékelő szűrőnélküli állapotában mérik.
- *Zaj*: Thermopile legjelentősebb zaja a Johnson zaj, mely az ellenállásának köszönhető. Értékét $\frac{nV}{\sqrt{Hz}}$ -ben adják meg.
- *Érzékenység*: Kimeneti feszültséget a hőmérséklet függvényében szokták megadni.
- *Látómező*: Függvényének rajzát szokták megadni, amelyből leolvasható, hogy a különbözőszögek alatt érkező sugarak mekkora erősséggel érzékelődnek.
- *Thermistor*: Thermopile-ok komoly hőmérsékletfüggését a tokba épített NTC thermistor felhasználásával szokták kezelni.

Három adattal jellemezhető:

- T_0 hőmérséklet, mely általában 25 °C
- R_0 thermistor ellenállása T_0 hőmérsékleten, tipikus érték 30 kΩ-100 kΩ között van.
- β paraméter

Ezek felhasználásával meghatározható a mért R -hez tartozó T hőmérséklet [19]:

$$T = \frac{\beta}{\ln \frac{R}{r_\infty}}, \text{ ahol } r_\infty$$

$$r_\infty = R_0 e^{-\frac{\beta}{T_0}}$$

Képletek NTC – re értendőek!

2.3.2.4 NDIR problémái és megoldásuk

Hőmérsékletet és nyomás: A csatornában a hőmérséklet és nyomás is változik. Ezeket folyamatosan mérni és korrigálni kell!

$$C = k * \frac{P * V}{T} [20]$$

C : koncentráció

k : konstans

P : nyomás

V : térfogat

T : hőmérséklet

Tehát, ha két mérési érték között megváltozik a hőmérséklet:

$$C_1 = k * \frac{P * V}{T_1}$$

$$C_2 = k * \frac{P * V}{T_2}$$

$$C_2 = C_1 * \frac{T_1}{T_2}$$

Tehát egy $\frac{T_2}{T_1}$ korrigálásra van szükség a mért eredményben.

Hasonlóan kapjuk meg a nyomást is:

$$C_1 = k * \frac{P_1 * V}{T}$$

$$C_2 = k * \frac{P_2 * V}{T}$$

$$C_2 = C_1 * \frac{P_2}{P_1}$$

Tehát egy $\frac{P_1}{P_2}$ korrekcióra van szükség a mért eredményben.

Nedvesség: Ergospiometriás mérésnél a kilehelt levegő rengeteg párát tartalmaz, mely a mérést befolyásolja, mert befolyásolja az infrafény terjedését és a készüléknek is rosszat tesz hosszabb távon. Ezekre megoldás, hogy a nyomás változtatásával kicsapatjuk a párát, egy nagyobb gyűjtött helyezünk el a szenzor előtt és/vagy nafion-ból készült csővezetékét használunk, mely átveszi a nedvességet. Mindkét módszer készen kapható alkarészt követel meg, így ezekkel nem kellett foglalkoznom.

Lámpa öregedése offszetet okoz. Erre egy megoldás, hogy gyakran kalibráljuk újra a készüléket, de erre sokszor nincs lehetőség, ezért egyszerűbb megoldás a lámpa

öregeedésének rögzítése és görbén ábrázolása. Minden egyes bekapcsolás után megmérjük a fényforrás erősségét és azt elmentjük, majd megfelelő időnként kiszámoljuk az így kapott értékekből az offszetet [21].

A por elleni védelem megoldható a szenzorhoz vezető csőbe helyezett 20 µm szivacsos szűrővel, amely megakadályozza a kosz továbbhaladását.

2.3.3 NDIR szenzor kalibrálása

Ahhoz, hogy az elkészített és megtervezett hardverünkkel mérni tudjunk, be kell tudnunk azt kalibrálni.

A kalibráció az alábbi három módon történhet:

- Kalibrációs gáz alkalmazásával,
- Légtéri levegő és nyugalmi állapotban kilélegzett levegő felhasználásával,
- Másik CO₂ eszköz igénybe vételével.

2.3.3.1 Kalibráció kalibrációs gáz alkalmazásával

Amikor a kalibrációt kalibrációs gázzal végezzük, akkor a legfontosabb számunkra, hogy olyan keverékeket használjunk, amelyek a mérési tartományunk határait pontosan lefedik, minél hosszabb ideig megtartja a gáz a pontosságát és felhasználhatósága egyszerűen.

Az első feltétel azt jelenti számunkra, hogy szükségünk van olyan keverékre, amellyel elő tudunk állítani 0%-os valamint 6-10%-os CO₂ koncentrációt.

A második feltételt a gyártó szabja, aki a kalibrációs gázt előállítja: mindig megjelölik egy felhasználhatósági intervallumot, mely időn belül tudják garantálni, hogy a palackból kiengedett gáz a megadott koncentrációs hibával fogja tartalmazni az adott gázokat.

Harmadik feltétel esetében törekedjünk, arra, hogy ha van választási lehetőségünk, akkor olyan gázt válasszunk, ami nem káros a szervezetre, minél kevésbé tűz- és robbanásveszélyes. Ez nagyban megnövelheti a gáz felhasználását. Ezeket az

adatokat az adott gáz biztonsági adatlapjából deríthetjük ki (MSDS – Material Safety Datasheet). Például az egyik felhasznált gáz lehet a százszázalékos oxigén. Ez a gáz nagyon gyúlékony és robbanásveszélyes, ezért biztonsági okokból csak állítva, kikötve szabad tárolni, hogy az eldőlését megakadályozzuk. Fontos még tudni, hogy ilyen koncentrációjú oxigén belélegzése hosszútávon roncsolja a tüdőt és hatással van az idegrendszerre.

Kalibráció menete kalibrációs gáz alkalmazásával: Kalibrációs gázt egy tartályba vezetjük, amelynek be- és kimeneti szelepét tudjuk zárni, illetve nyitni. A gázt elkezdjük beleengedni a tartályba és közben a tartályba helyezett hőmérővel és nyomásmérővel mérjük a hőmérsékletet és nyomást. Ez arra jó, hogy vissza tudjuk ellenőrizni, hogy a tartályban lévő levegő és a beleengedett keverék milyen új elegyet hoz létre. Ha a tartályban a számunkra megfelelő mennyiségű gáz van, akkor elzárjuk a bemeneti szelepet. Végzünk egy végső mérést és abból becsüljük, hogy milyen a keverék koncentrációja. Ezek után ezt a keveréket átengedjük a szenzorunkon és megvárjuk azt a pillanatot, amikor a tartályban lévő nyomás eléri a légkörit. Ekkor lesz a szenzorban mért érték a keverékkel megegyező. A mérés pontatlansága miatt érdemes több pont átlagával számolni. Általában öt pontra szoktak mozgóátlagolást végezni [22].

2.3.3.2 Kalibráció légtéri levegő és nyugalmi állapotban kielégített levegő felhasználásával

Előzetes mérések elvégzéséhez nem szükséges teljesen pontosan kalibrált eszköz. Ehhez elegendő, ha a kalibrálás is csak kezdetleges. Ekkor feltevésekkel kalibrálhatjuk be az eszközt. Ha egy jól kiszellőztetett szobában vagyunk vagy szabadtéren, akkor a hardver által mért értéket 0,3%-ra becsülhetjük. Másik pontot nyugalmi esetben az emberi szervezet által kibocsátott széndioxid szint adja, ami körülbelül 4%.

2.3.3.3 Kalibráció másik CO₂ eszköz igénybevételével

Harmadik lehetőség a kalibrációra, hogy a kalibrálandó eszközt egy már kalibrált eszközzel kötjük sorba. Erre például megfelelő eszköz a Goldwei által forgalmazott C200 modul. Nagy előnye, hogy saját vákuumpumpával rendelkezik, így

vékony mintavevő csővel sorba köthető a kalibrálandó eszközzel, és a kalibrálandó eszköz bemenetére különböző tartalmú CO₂-ot juttatva megmérhetjük, hogy különböző pontokon milyen értéket mér a szenzorunk. Ennek nagy előnye, hogy nincs szükség kalibrációs gázra, és egy ki- és belégzés alatt széles értéktartományt bocsátunk az érzékelőkre, emiatt több mint két ponton kalibráljuk össze az eszközöket.

Itt elég fontos, hogy valahogyan meg kell tudnunk, hogy a mért jelek milyen késleltetést szenvednek, hiszen a második érzékelő később kapja meg ugyanazt a gázt. Ennek következtében szükségünk van egy olyan mérési eljárásra is, amikor ezt a késleltetést határozzuk meg. Erre a legegyszerűbb megoldás, ha elkezdjük áramoltatni a két eszközön keresztül a légköri levegőt például egy elektromos pumával. Amikor az érzékelők stabilan beállnak egy-egy értékre, akkor belefújunk és vizsgáljuk, hogy mikor kezdenek az értékek eltérni az eddigi állandó értékektől. Ha az eredményeket egy grafikonon ábrázolhatjuk, akkor a késleltetés könnyen azonosítható és ezután a két görbe egymásba fésülhető [23].

2.3.4 Szén-dioxid szenzor további felhasználása

Szén-dioxid szenzorokat ergospirometrián kívül használnak még kapnográfokban és spirometriai eszközök kalibrálására is.

Kapnográfok feladata lélegeztetés során ki- és beáramló gáz szén-dioxid-tartalmának folyamatos mérése. A kapnográf görbét is rajzol, mely hasonló egy nyomásgörbéhez, míg a kapnométer csak az értékeket jeleníti meg. Mindkét készülék képes a légzési frekvencia figyelésére is.

Kapnográfot intubáció után használják két célból is:

- Segíti megállapítani a tubus helyzetét (légcsőbe vezetett tubus esetén van nyelőcsőbe vezetett esetén nincs CO₂- kiáramlás.
- Segíti a lélegeztetés frekvenciájának és térfogatának a beállítását (Alullélegeztetett betegnél nő, túllélegeztetett betegnél csökken a CO₂-kiáramlás) [24][25].

Ultrahangos spirométerek kalibrációja és használata során, a készüléken keresztüláramló széndioxidot közelítő értékekkel korrigálják: 0,04% beáramló és 4,04%

kiáramló. Mind a kalibráció, mind a mérés során használt széndioxid szenzor javítaná a pontosságot.

3 Részletes specifikáció

Az előző fejezetben összegyűjtött információk együttesen meghatározzák a rendszerünkkel szemben támasztott elvárásokat, amely lehetővé teszi a specifikáció elkészítését.

3.1 A rendszer részletes specifikációja

Hardver specifikációja:

- Legfontosabb, hogy rendelkezzen analóg/digitális átalakítóval, hogy a szenzorok (thermistor, nyomás, kalibráló eszköz, és thermopile) analóg jeleit mérni tudja.
- Thermopile és thermistor jelei alacsonyak, így azokat erősíteni és szűrni kell.
- Az eszköznek képesnek kell lennie kommunikálni a PC – vel.hogy a mért adatokat elküldje, valamint vezérlő utasításokat megkapja, ezért szükség van USB (Universal Serial Bus) vezérlésre alkalmas perifériára.
- A mérést befolyásoló tényezők mérése miatt, szükség van a tok hőmérsékletét valamint a csatornában lévő nyomás meghatározására alkalmas szenzorokra.
- A csatornába be kell juttatni a mérni kívánt összetételt, amely egy vákuumpumpával könnyen elérhető.
- Kalibrációhoz felhasznált eszköznek tápellátásra lesz szüksége, valamint annak jeleit mérni, esetleges vezérlő parancsait kezelnie kell.
- Lámpa öregedését megelőzendő az eszköznek képesnek kell lennie kikapcsolnia azt, amikor nincs használatban.

PC szoftver specifikációja:

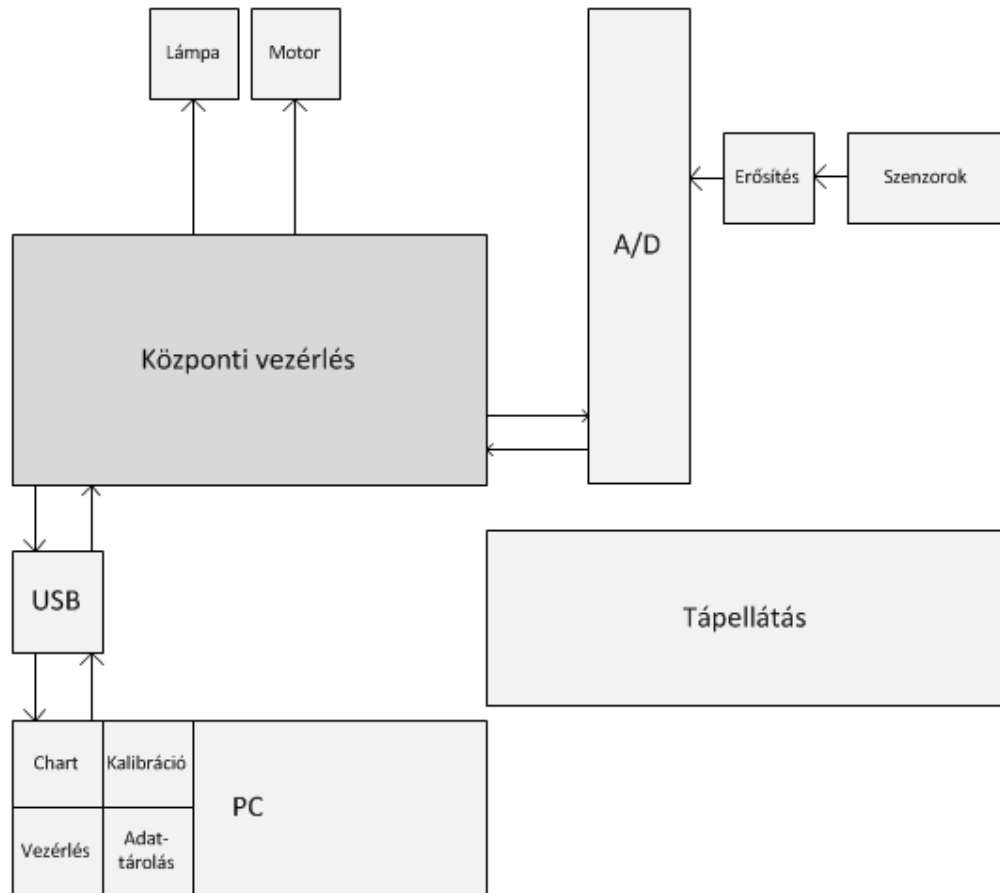
- USB vonalon érkező nyers adatok összegyűjtése, feldolgozása, tárolása és dinamikus megjelenítése.

- Az eszköz számára parancs utasítások küldése.
- Kalibrációs adatok meghatározása, tárolása és ellenőrzés végett grafikus megjelenítése.
- Nyers és/vagy feldolgozott adatok exportálása.

4 Rendszerterv

4.1 Logikai rendszerterv

Előző fejezetben felvázolt specifikáció alapján a logikai rendszerterv vázlata:



4.1. ábra: Logikai rendszerterv

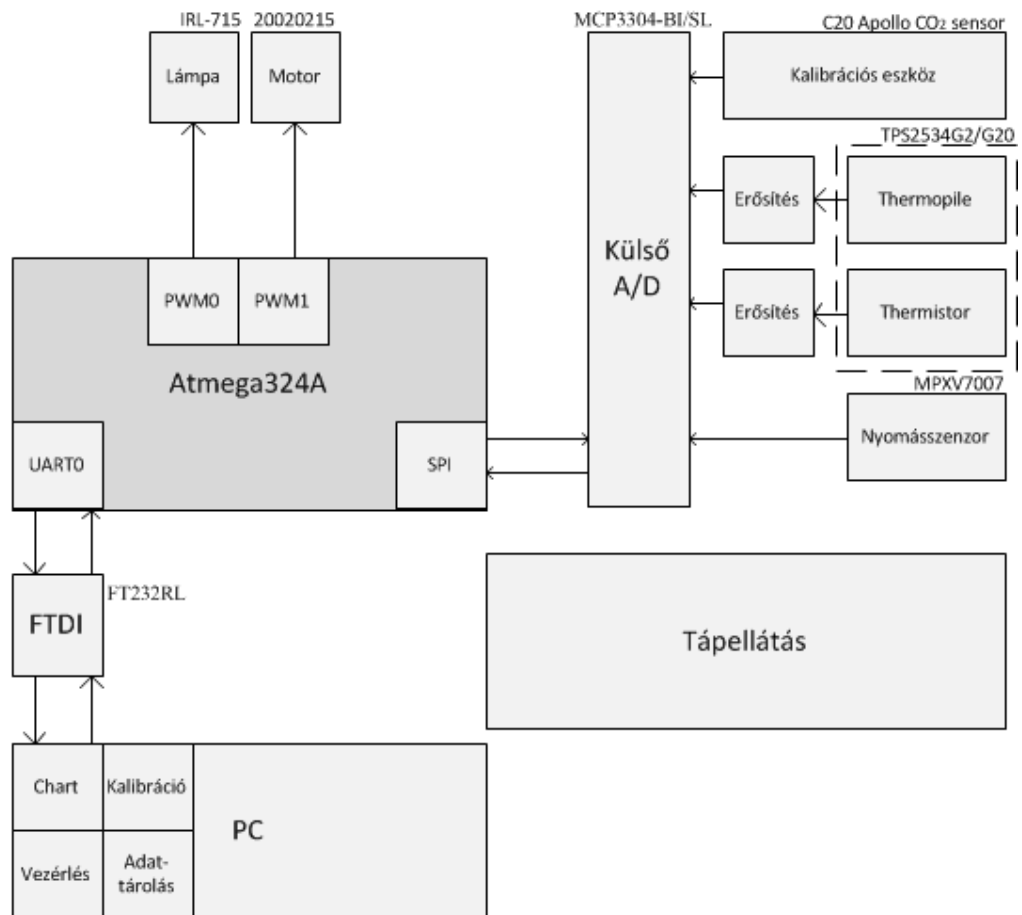
A központi vezérlés feladata a perifériák és I/O lábak inicializálása, a PC-től érkező parancsok figyelése és az A/D vezérlése. A PC-től kapott vezérlő parancsok alapján különböző működési állapotokat kell beállítania és támogatnia. Legfőbb feladata a mérési adatok begyűjtése az A/D-től és azok továbbítása a PC felé. További feladata még a mérést lehetővé tevő lámpa és pompa vezérlése.

A PC-n futó szoftver az USB vonalon érkező nyers adatok összegyűjti, feldolgozza, tárolja és dinamikusán megjeleníti. Feladata még a felhasználó által indított

kérések továbbítása a hardver számára, valamint a kalibrációs folyamat vezérlése, és a kalibrációs adatok meghatározása, eltárolása.

4.2 Hardverterv

A logikai rendszerterv alapján elkészíthető rendszer hardverterve majd pedig a kapcsolási rajz. A kiválasztott integrált áramkörökkel kiegészített rendszerterv az alábbi ábrán (4.2. ábra) látható.



4.2. ábra: Hardverterv

4.2.1 Mikrokontroller

A központi vezérlőegység kiválasztásánál a legfontosabb szempontok a feladat teljesítményigénye és a felhasználni tervezett perifériák. Ezeken kívül érdemes még

figyelembe venni a vezérlő programozási lehetőségeit: milyen elérhető fejlesztő környezete van, valamint támogat-e valamilyen magas szintű programozási nyelvet. A fejlesztésre szánt időjelentősen csökkenthető egy magasszintű nyelv támogatása esetén, de mivel egy fordító segítségével állítjuk elő az Assembly kódot, így a programunk futási sebessége lassabb lesz. A sebességkritikus részeket ezért Assembly nyelven szokás megírni és azokat a C kódba behelyezni.

A mikrokontrollerrel szemben támasztott elvárásaink alapján az alábbi perifériákkal kell rendelkeznie:

- *UART*: Az FTDI chippel való kommunikációhoz van szükség erre, mely az USB vezérlésért felelős.
- *SPI*: A külső A/D –vel való kommunikációhoz kell.
- *PWM*: A pumpa teljesítményének rugalmas beállításához.

Ez a három periféria csak kis mértékben szűkíti le a piacot, mert majdnem valamennyi mikrokontroller rendelkezik ezekkel. Az Egyetemi tanulmányaim során többször is volt alkalmam az Atmel nyolcbites mikrokontrollereivel dolgoznom, amelyek során rengeteg pozitív élmény ért: rendkívül jól dokumentált felhasználói kézikönyv, elérhető ingyenes fejlesztőkörnyezet, valamint az előre megírt rengeteg header fájl egyértelművé tette számomra a választást. Ezért ezen család tagjai közül az alábbi szempontok alapján választottam:

- könnyen be lehessen szerezni,
- olyan tokkal rendelkezzen, melyet kézzel be lehet forrasztani,
- korábban vásárolt AVRISP mkII-vel programozható legyen,
- tartalmazza a fent említett perifériákat,
- valamint a későbbiekben tervezett továbbfejlesztéshez ne kelljen új kontrollert választani.

Így esett a választásom az **Atmega324A**-ra, melynek az alábbi rövid felsorolás szemlélteti legfontosabb tulajdonságait:

- Továbbfejlesztett RISC architektúra
- 131 főként egy órajelciklus alatt végrehajtódó parancs

- 32 darab 8 bites általános célú regiszter
- Maximális 20 MIPS 20 MHz-en
- Kétciklusos hardveres szorzó
- 32 kbyte-os Flash (10 000 írás/törlés ciklus)
- 1 kbyte-os EEPROM (10 000 írás/törlés ciklus)
- 2 kbyte-os belső SRAM
- Bootloader támogatás, saját védettséget biztosító bittel
- JTAG interfész
- Perifériák:
 - Kettő 8 bites időzítő
 - Egy 16 bites időzítő
 - 6 darab PWM csatorna
 - 8 csatornás 10 bites ADC
 - I²C
 - SPI
 - Kettő USART
 - Külső/belső megszakítók támogatása
 - Hat fogyasztás csökkentet állapot
- Működési feszültségtartomány: 1,8-5,5 V
 - 0-4 MHz @ 1.8-5.5 V
 - 0-10 MHz @ 2.7-5.5 V
 - 0-20 MHz @ 4.5-5.5 V
- Fogyasztás
 - Aktív állapotban: 0.4 mA
 - „Power-down” módban: 0.1 μA
 - „Power-save” módban: 0.6 μA [26].

4.2.2 Külső A/D

Az Atmega324A rendelkezik 8 csatornás 10 bites A/D-vel, de a 10 bit számunkra nem megfelelő, főleg, ha figyelembe vesszük a $\pm 2\text{LSB}$ -nyi pontatlanságot is, amely 5 V-os referenciafeszültség esetén $\pm 8\text{ mV}$ -nak felel meg. Egyéb paraméterei alapján megfelelő lenne a feladatnak.

Ezért egy külső A/D-re volt szükség, amely minimálisan 12 bites, INL és DNL egyike sem nagyobb $\pm 1\text{ LSB}$ -nél, és rendelkezik öt csatornával. A választás a Microchip által gyártott **MCP3304** – re esett.

Tulajdonságai:

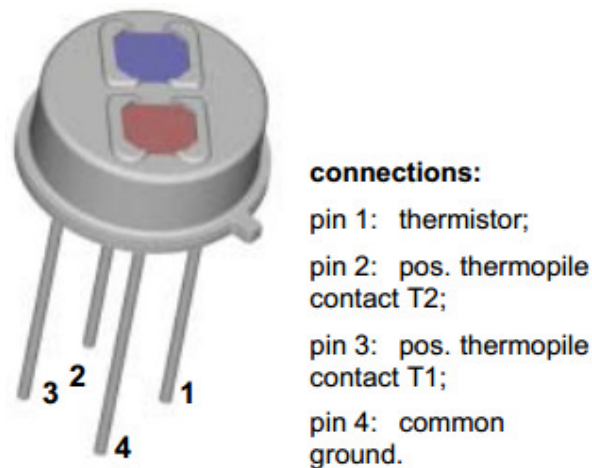
- Bemenetei:
 - 4 darab differenciális vagy
 - 8 darab egycsatornás
- Maximum $\pm 1\text{ LSB}$ DNL
- Maximum $\pm 1\text{ LSB}$ maximum INL
- 13 bites
- SPI interfész
- Tápellátás: 2,7 V-5 V;
- Mintavételi sebesség
 - 100 ksps mintavételezés 5 V-on.
 - 40 ksps mintavételezés 2,7 V-on.
- Standby állapotban 50 nA, míg 5 V tápellátás esetén aktív állapotban 450 uA fogyaszt.
- 16 lábás SOIC tok [27].

4.2.3 Szenzor és lámpa

NDIR módszer bemutatásánál részletesen felvázoltam a működés alapját és a szenzorral szemben támasztott elvárásokat. A legegyszerűbb és a mérés szempontjából

legoptimálisabb csatornát abban az esetben kapunk, ha egyetlen tokban van két darab thermopileunk. Ilyen az általam is használt **TPS 2534 G2 G20**.

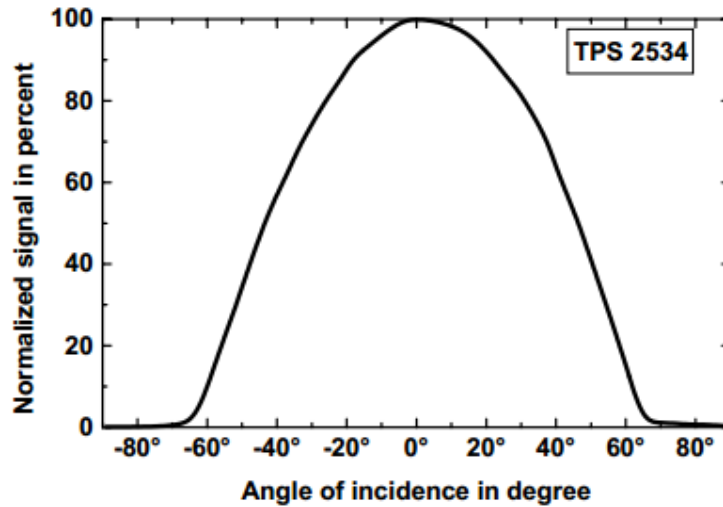
A TPS2534 két thermopilet (G20 és G2) és egy termisztort tartalmaz egyetlen TO 5 tokban. A G20–ast referencia, míg a G2-t CO₂ szelektív szűrővel látták el. A közös toknak köszönhetően a mérés során biztosítva van a két thermopile azonos hőmérséklete, valamint a termisztornak köszönhetően a hőmérsékletváltozás is nyomon követhető, és a szükséges korrekciók elvégezhetőek.



4.3. ábra: TPS2534 szenzor felépítése. Forrás: [22]

Tulajdonságai:

- Besugárzott infravörös fény hatására feszültségforrásként működik
- A referenciaérzékelő előtt 4.0 μm -es, míg a CO₂ szelektív szűrővel ellátott érzékelő előtt 4.26 μm -es középhullámhosszú szűrő található.
- Thermistor adatai:
 - $R_0 \sim 30 \text{ k}\Omega$
 - $T_0 \sim 25 \text{ }^\circ\text{C}$
 - $B \sim 3964 \text{ K}$
- Látómező 90°



4.4. ábra: TPS2534 látómezeje. Forrás: [22]

Fényforrásként a gyártó által ajánlott és készített **IRL 715** lett beépítve a szenzorba.

Adatai:

- Tápfeszültség 5 V
- Teljesítmény 575 mW
- Bemeneti áram 115 mA [22].

4.2.4 Kalibrációs szenzor

Kalibrációs célra az Apollo Union által készített **C2010540N**, mellyel korábbi munkám során pozitív tapasztalatokat szereztem. Nagyban megkönnyíti felhasználhatóságát, hogy kalibrációja, egyszerűen elvégezhető: CO₂-t nem tartalmazó környezetben a rajta lévő gomb lenyomásával nullázható. Fontos, hogy mintavételi csatornájába diffúzióval áramlik be a gáz, ebből adódik lassú reakcióideje.



4.5. ábra: C20 szenzor. Forrás: [28]

Tulajdonságai:

- Tápellátás: 5 V (4,7-5,5 V, 240 mA áramcsúccsal)
- Lineáris kimenet: 0,5 V-5 V
- 0-10% széndioxid érzékelésére alkalmas
- Pontosság: ± 50 ppm +/- 5% olvasási
- Nyomástól függ: 0,13%/mmHg
- Válaszidő < 4 másodperc [28].

4.2.5 Nyomásszenzor

Nyomásszenzort a már említett korrekciós szerepén kívül, a csatornában légzés miatt ingadozó nyomásfigyelésre is alkalmazható, így pontos ki-belégzési idő mérhető, mely fontos spirometriai érték.

Nyomásszenzornak a **MPXV7007DP** választottam, amely -7 és 7 kPa között differenciális bemenetein méri a nyomáskülönbséget. Kisebb változatra -2 és 2 kPa mérésére alkalmas, de mivel az emberi kilégzés maximuma közel a 2 kPa körül mozog, valamint kalibrációs gáz alkalmazásánál is általában nagyobb nyomás alakul ki, ezért célszerű ezt használni. Egy bemenetű változatával szemben a két bemenet lehetővé teszi, hogy mindig a külső nyomáshoz lehessen meghatározni a csatornában lévő nyomást.

Tulajdonságai:

- Tápellátás 5 V (4,75-5,25 V)
- Áramfelvétel 7 mA (max. 10 mA)
- Kimenete lineáris: 0-5 V között, ahol a ~2.5 V a 0 kPa-nak megfelelő érték.
- Hőmérsékletkompenzált: 0-85 °C mentén maximum 5% hiba adódhat [29].

4.2.6 Pumpa

Pumpának a Thomson által gyártott 2002VDLC típusú eszközt választottam. Ez 500 ml/min maximális áramlás és 400 mbar nyomás előállítására képes. Tápellátása 3,5 V és 5 V között változtatható, amellyel egyben az áramlási sebességet is változtatjuk ~ 320 ml/min és a már fent említett érték között. Maximális áramfelvétele 90 mA.

5 Hardvertervezés

5.1 Fejlesztői környezet

Hardvertervezés során meg kell tervezni a kapcsolási rajzot, az abban felhasznált valamennyi alkatrész beültetési rajzát elkészíteni, majd elhelyezni a panelen, és a kapcsolási rajznak megfelelően összekötni. Az áramkör tervezésére az Altium Designer nevű programot használtam, mely sok beépített funkciójával támogatja az előzőekben leírt lépéseket.

5.2 Kapcsolási rajz

A kapcsolási rajznál érdemes arra törekedni, hogy könnyen átláthatóak legyenek az egyes oldalak, mindegyikre egy-egy funkcionálisan összetartozó rész tervezni. Az alkatrészek kiválasztásánál a betöltendő funkció mellett a kiválasztást befolyásolta az adott alkatrész beszerezhetősége, ára és mérete, mely a kézzel történő beforrasztás miatt volt fontos. A szoftver egy-egy alkatrészt egyetlen egységbe foglal, tárolja kapcsolásirajz-szimbólumának, beültetési rajzának, valamint rengeteg egyéb paraméter is megadható: rendelési szám, ár, paramétereinek érték stb.

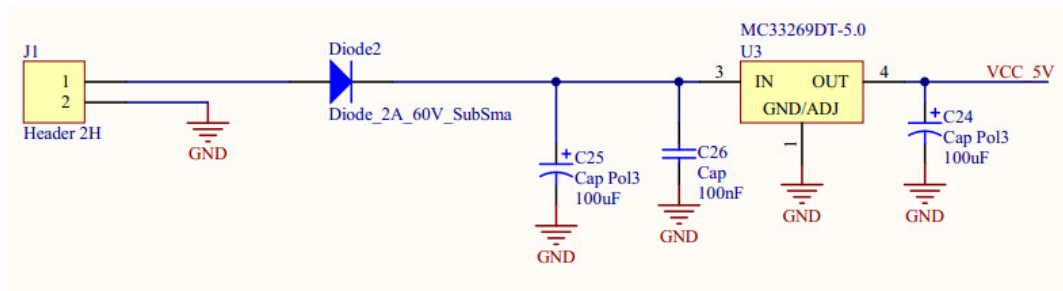
Valamennyi tápellátást igénylő alkatrész mellett egy 100 nF-os kondenzátor is el lett helyezve, amelynek feladata a nagyfrekvenciás zajok kialakulásának megakadályozása és elnyomásának segítése.

A teljes kapcsolási rajz a függelékben megtalálható, a következőkben csak a lényegesebb elemek tárgyalására térek ki.

5.2.1 Tápellátás

Első tervek alapján az USB port egyben a tápellátás forrása is lett volna, de mivel a szabvány széles tartományt enged meg a maximálisan elérhető áramszintje, valamint az előzetes számítások alapján a teljes panel tápfelvétele meghaladja a legnagyobb terhelés esetén a 400 mA-t. A panelen lévő alkatrészek ellátása 5 V-ról történik, amelyet egy zajos külső 12 V-os adapterből állítódik elő. Nincs szükség egyéb

feszültszint elállítására, mert valamennyi részegységnek megfelelő az 5 V. A tápforrás bemeneténél védődíoda lett elhelyezve az esetlegesen előforduló fordított polaritású adapter elleni védelem miatt.



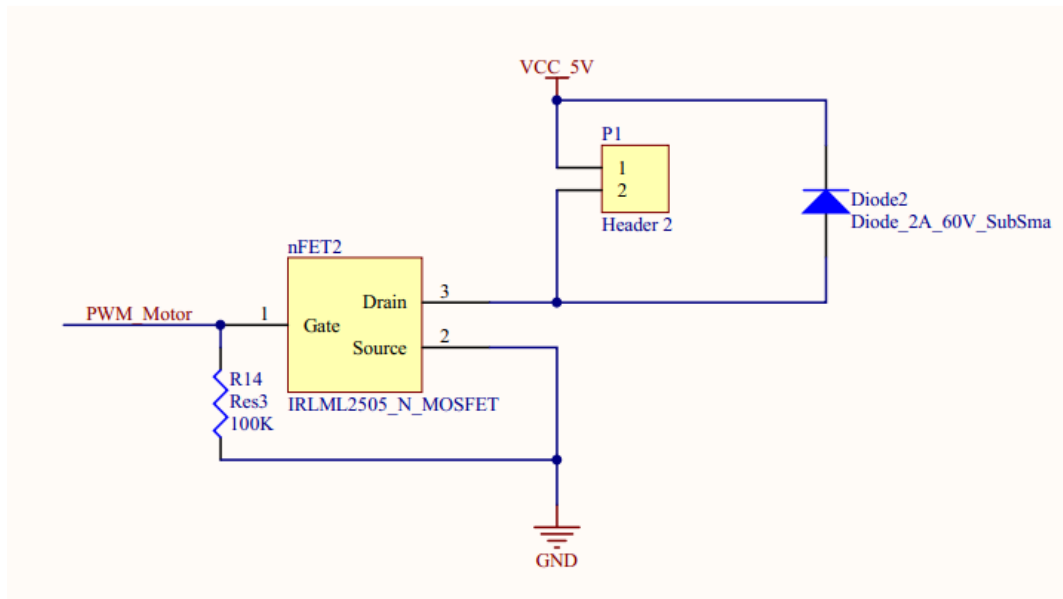
5.1. ábra: Tápellátás

5.2.2 Mikrokontroller

A feladata megoldásához egy Atmega324A lett felhasználva. Órajeléért egy külső 20 MHz-es oszcillátor felelős, mely 22 pF kondenzátorokon keresztül van földelve. A reset láb egy 10 k Ω ellenálláson keresztül 5 V-ra van felhúzva, ezzel biztosítva a magas szintet a bementén. Zavarérzékenységet 100 nF-os kondenzátor biztosítja. Fontos megjegyezni, hogy az a reset lábon elhelyezett kondenzátor miatt, bizonyos programozók nem működnek, ez jelen esetben nem állt fent.

Az AVRISP MKII programozónak a MOSI, MISO, SCK, RESET, VCC, GND jelekre van szüksége, hogy a mikrokontrollert későbbiekben beforrasztott állapotban is programozni tudja. Ehhez kábelének megfelelő port lett kialakítva. A MOSI, MISO és SCK lábakat az SPI periféria is használja, ezért, hogy az eszköz programozható lehessen 1 k Ω ellenállások lettek elhelyezve a külső A/D felé vezető vezetékéken.

Mikrokontroller feladata még a lámpa és a pumpa vezérlése, ki-bekapcsolása a már említett okok miatt. Mindkettőt egy PWM kimenetként is funkcionáló portra lett tervezve, mellyel szabályozható szívteljesítmény, valamint a lámpánál a modulált fényforrás hatása is kielemezhető. A kapcsolgatáshoz egy IRLML2505 NFET-et alkalmaztam, melynek legfontosabb paramétere, hogy bekapcsolt állapotban a drain-source között kialakuló ellenállás 0,045 Ω . Ezen így nagyon kis feszültség esik, hiszen a vele sorba kötött lámpa és pumpa k Ω nagyságrendű ellenállással rendelkeznek. A kikapcsolást segítő egy 100 k Ω ellenállás lett tervezve a gate-re.



5.2. ábra: Pumpa vezérlése

5.2.3 Szenzorok jeleinek erősítése

A két thermopile, valamint a termisztor jelei erősítést igényelnek. A kapcsolást a gyártó által javasoltak alapján terveztem meg.

Az 1-es 4-es lábak között található a termisztor, mely a rajta átfolyó áram hatására melegszik, így meghamisíthatja a mérést, ezért fontos a rajta átfolyó áram ne legyen nagyobb $5 \mu\text{A}$ -nél. Normális használat esetén a mérés közben a termisztor ellenállása $10 \text{ k}\Omega$ (50°C) és $100 \text{ k}\Omega$ (0°C) intervallumon kívül biztosan nem fog esni, így a feltétel az R_v értékének $1 \text{ M}\Omega$ -ra választásával biztosítható.

Az erősítőre is tesz ajánlást. Legyen alacsony zajú, valamint abban az esetben, ha nem rail-to-rail, akkor +/- tápellátás használata javasolt. A követelmények tökéletesen megfelel a MicroChip által gyártott MCP6V02-E/SN erősítő, mely mind orvosi készülékekben, mind thermopileos kapcsolásokban is elterjedten alkalmazott. Legfontosabb tulajdonságai:

- Beépített automatikus offszet nullázó áramkör,
- Egy polaritású táp: 2.7 V - 5.5 V
- Rail-to-rail kimenet
- Sávszélessége 2.8 MHz

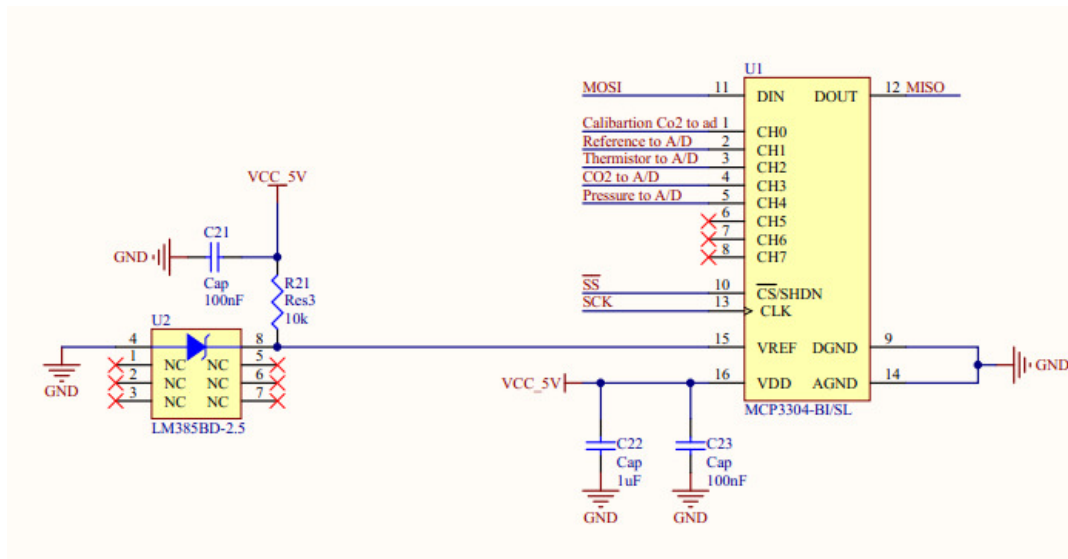
A javasolt kapcsoláshoz kétpólusú aluláteresztő szűrővel lett kiegészítve, melynek vágási frekvenciája 30Hz, hogy a kétszeresen túlmintavételezett thermopile-ok jelein az aliasing-ot meggátoljuk.

5.2.4 Külső A/D

Külső A/D-nek a már bemutatott **MCP3304** nyolc csatornás, 13 bites átalakítót választottam. Ez elegendő bemenettel rendelkezik, így nem volt szükség egyáltalán a mikrokontrollerben található belső átalakító használatára.

Működéséhez elengedhetetlen a referencia feszültségének pontos beállítása, melyhez **LM385D-2.5**-ös referenciadiódát használtam fel, melynek pontossága +/- 1.5%. Fontos, hogy a rajta átfolyó áram nem haladhatja meg a 20 mA, ezért a katódláb 10 k Ω -on keresztül van 5 V-ra felhúzza.

A referencia feszültség 2.5 V választása miatt a nyomás, valamint a külső CO₂ szenzor kimenetét feszültségosztóval felezni kell.



5.3. ábra: Külső AD

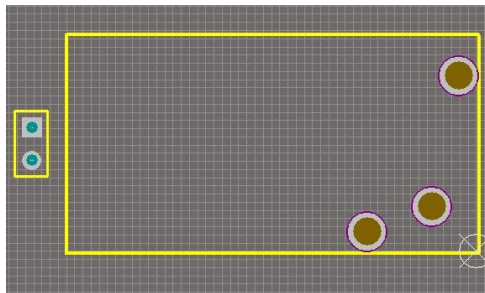
5.3 Nyomtatott áramkör elkészítése

5.3.1 Beültetési rajzok (footprintek)

Mind már szó volt róla, az Altium Designer legfőbb előnye, hogy rengeteg alkatrész készen megtalálható a programban, valamint mások által készítették is hozzáadhatóak a projekthez.

Általában, azért egy-egy alkatrészt magunknak kell elkészíteni. Ekkor két esett lehet, hogy a beültetési rajz valamilyen hagyományos tok és azt nem kell nekünk megrajzolni. Ekkor az alkatrész sematikus rajzának elkészítése után annak láb kiosztást kell összeegyeztetni a beültetési rajz lábaival. Én esetemben ilyen volt az Atmega324, melynek tokozása TQFP44_N, amely készen volt már, de kapcsolási rajzjele hiányzott.

Másik eset az, amikor az alkatrész beültetési rajza sem található meg, vagy csak részben található meg. Ilyen volt a pumpa és a csatorna is. Ezeknél fontos volt, hogy mechanikailag kiterjedt alkatrész mindenképpen a hozzátartozó csatlakozó(k) mellé legyen elhelyezve, valamint rögzítési pontjai is megfelelőek legyenek.



5.4. ábra: Pumpa beültetési rajza

5.3.2 Szabályok beállítása

Az áramkör igényeinek teljesen mértékben megfelel egy kétrétegű nyomtatott áramkör. Az alsó réteg nagy része összefüggő földpotenciál, csak nagyon szükséges esetben lett átvezetve vián egy-egy jelvezeték rá, akkor is minimális távon. A nyáklemez alapanyaga az általánosan használt FR4 üvegszövettel erősített epoxi vázas 18-35 mikronos rézzel foliozott 1,5 mm vastag lemez [30].

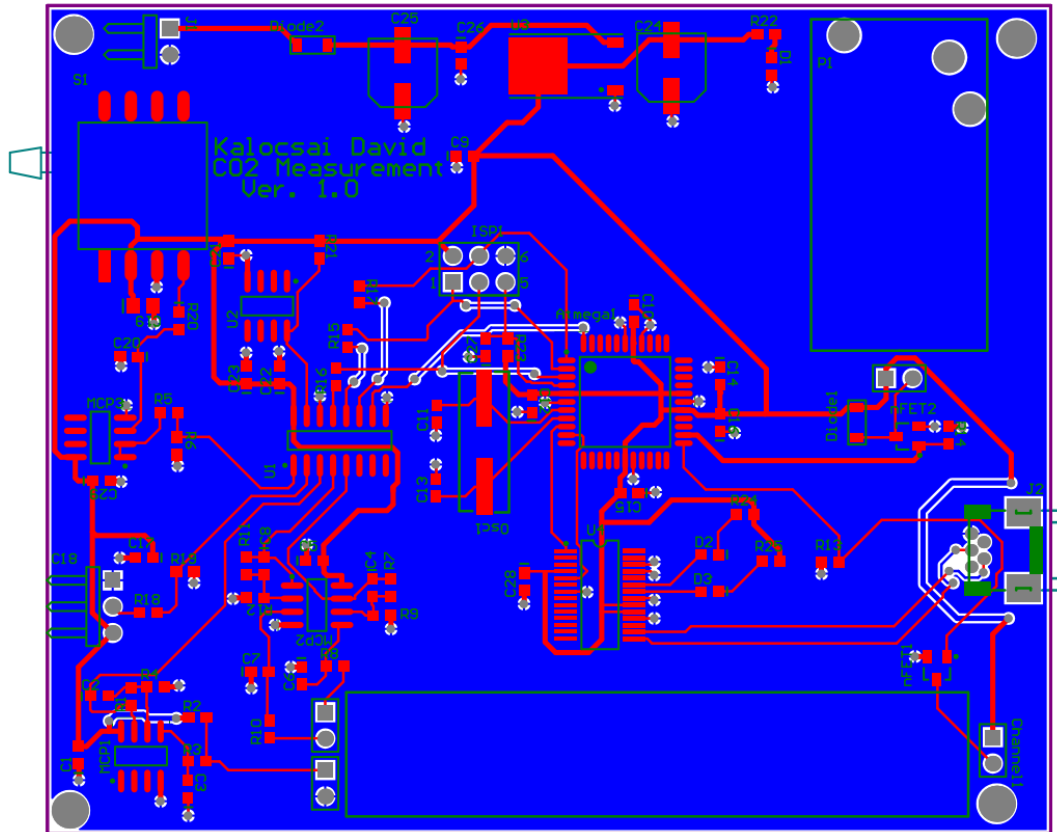
A tervezés során a gyártásra vonatkozó határértékek betartása kötelező, melyeket a gyártó határoz meg. Satronik által megkívánt határértékek:

- Legkisebb gyártható nyákméret: 10 cm x 16 cm, kisebb nyákok esetén célszerű tehát többet összemontírozni.
- Legnagyobb nyákméret: 30 cm x 45 cm.
- Legkisebb vezeték szélesség: 0,254 mm.
- Legkisebb szigetelési távolság: 0,2032 mm
- Legkisebb furat: 0,5 mm, de javasolt a 0.7 mm vagy nagyobb, ha a méretek megengedik
- Legnagyobb furat: NC géppel maximum 4,9 mm furat fúrható [30].

Az Altium Designer ezekben is segítséget nyújt, valamennyi határérték előre beállítható, és ezek megszegése esetén figyelmeztet minket a szoftver a fordítás során.

5.3.3 Alkatrészek elhelyezése

Az alkatrészek elhelyezése az egyik legfontosabb és egyben legnehezebb része a nyáktervezésnek. Fontos, hogy alapvető szabályokkal tisztában legyünk és azokat maximálisan betartsuk, mely a későbbi felhasználás során rengeteg problémától szabadíthat meg minket.



5.5. ábra: Panel PCB rajza

A panelomat az indokoltnál nagyobbra terveztem, amely a kalibráció során praktikus, valamint későbbiekre tervezett kiegészítők miatt van szerepe (ezek nem része a diplomamunkámnak). A tápellátás a panel tetején foglal helyet, mely a tiszta 5 V előállítás után kettéágazik és így látja el a nyák két ellentétes oldalán elhelyezkedő digitális és analóg részt. A nyák közepén található a mikrokontroller, melytől balra vannak az analóg alkatrészek: a thermopile-ok, a termisztor, azok erősítői, valamint a nyomásszenzor és a külső A/D. Az A/D is úgy lett lehelyezve, hogy analóg bemenetei legyenek az analóg rész felé. Míg az ellenkező oldalon található az USB, a lámpa és a pumpa. A mikrokontroller felett található a programozó port. Ez a kettéválasztás az analóg vezetéseket védi a nagyfrekvenciás digitális vezetésektől származó zajoktól.

5.3.4 Vezetékezés

Vezetékezés az elfogadott alapszabályok betartása mellett történt:

- Tápvezeték megfelelő vastagságának betartása, melyet ahol csak lehet.(5 V esetén 0.5 mm)
- Többi jelvezeték vastagsága: 0,254 mm.
- A vezetékekben kerülni kell a derékszögű töréseket, mert ezek a pontok reflexió forrásai lehetnek, valamint a gyártás során a maratót megnehezítik.
- Egymáshoz csatlakozó rézfelületeknél a hegyes szögek kerülése ajánlott, mert itt is érvényesül a csúcshatás: a nyílásszög csökkenésével növekszik a kialakuló térerő.

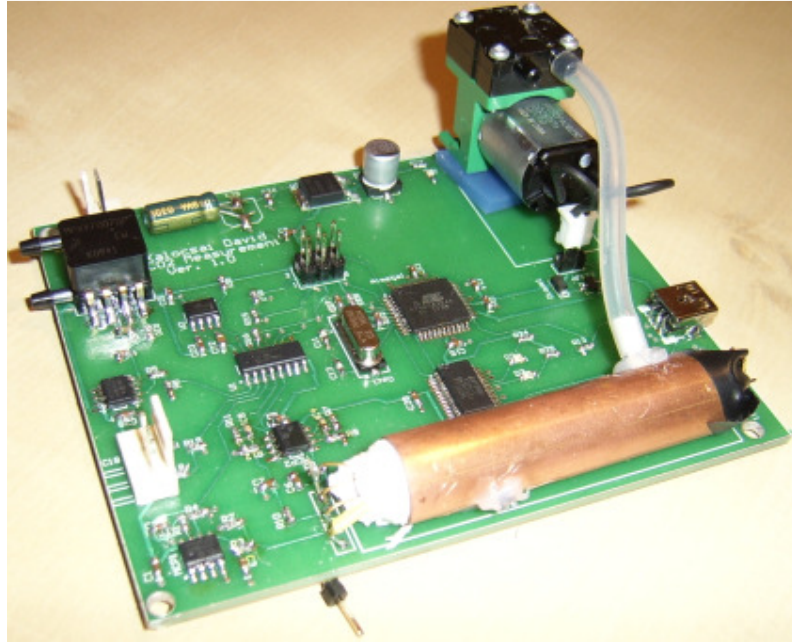
5.3.5 Gyártófájlok elkészítése

Ezzel a hardverterv teljessé vált. Következhet a nyomtatott áramkört gyártása, melyhez elő kell állítani a szükséges fájlokat. Az Altium Designer –ben, amely legenerálja az NC drill és Gerber fájlokat nekünk.

5.3.6 Élesztés

Az élesztést a tápellátás beültetésével érdemes kezdeni, hogy az esetlegesen felmerülő hibás feszültség szint vagy áram ne okozzon kárt semmilyen alkatrészben. Ennek ellenőrzése után következhetett a mikrokontroller, valamint az alaplátához szükséges alkatrészek beforsztása. Ezután az AVRISP MKII-vel megpróbáltam felprogramozni az eszközt, de nem jártam sikerrel. A problémát a programozó órajele okozta, mely nagyobb volt, mint a mikrokontroller órajelének $\frac{1}{4}$ -e. Ennek megszüntetése után sikerült kommunikálni a mikrokontrollerrel.

Ezek után következtek az analóg részek, majd a digitális oldal, valamennyi egységet egyesével ellenőrizve. A pumpa és a szenzor ezután lett elhelyezve a panelen (5.6 ábra).



5.6. ábra: Kész hardver képe

6 Firmware

A hardver felélesztése után következhetett az Atmega324 szoftverének megírása. Fejlesztői környezetnek az Atmel által készített, legújabb verziója Visual Studio alapokon futó Atmel Studio használható, melynek rengeteg, kényelmes funkciója segíti a fejlesztést. Egyik leghasznosabb funkció a beépített szimulátor, melynek segítségével a program bármely pillanatában leállítható, ezután lekérdezhetőek a regiszterek és kimenetek állapota, valamint azok szabadon megváltoztathatóak. Ugyancsak képes a .hex és .elf fájlok előállítására, feltöltésére és a mikrokontroller főbb paramétereinek felprogramozására a megfelelő programozó segítségével.

6.1 Firmware alapjai

6.1.1 Állapotok

A szoftver központi része egy állapotgép, mely az alábbi állapotokat ismeri:

- „*No Measurement*”: Ekkor a készülék időzítő hatására megnézi, hogy kell-e állapotot váltani. Egyéb feladatot nem végez. Későbbiekben ekkor érdemes lehet egy teljesítménycsökkentett mód beállítása.
- „*Warm Up*”: Ebben az állapotban bekapcsolja a lámpát, majd 60Hz-zel méri az érzékelő hőmérsékleti adatait és azt továbbítja a számítógép számára. Ekkor egy exponenciálisan beálló görbét kapunk.
- „*Measurement*”: Ez felel meg a későbbiekben használt széndioxid szint mérésnek. Ekkor bekapcsolja a lámpát és a motort is, majd 60Hz-zel mintavételezi a thermopile-okat, termisztort és nyomást és elküldi a mért adatokat a PC-nek.
- „*Calibration*”: Ebben az állapotban történik a széndioxid bekalibrálása, a „*Measurement*” állapottól abban különbözik, hogy a külső kalibrációs széndioxid értékét is méri és továbbítja.

6.1.2 Firmware egységei

A firmware logikailag az alábbi egységekre bontható le:

- Main: Itt található a központi vezérlésért felelős végtelen ciklusban futó állapotgép, mely minden ciklus után vár 20 us-ot. Ha adat érkezik az USB-ről, akkor megszakítás generálódik. A megszakítási függvény az érkezett karakternek megfelelő státuszváltozókat beállítja. Háromféle státuszváltozó létezik: motor, lamp, és measurement. Ezeket a soros porton érkező 'l', 'p', 'm', 'c', 'w', 'o' (lamp, pump, measurement, warm up, off), parancsok segítségével lehet állítani. A megszakítás függvényben csak a változó értéke állítódik a tényleges végrehatás a főciklus feladata. Parancsok rövid lényege:
 - 'l' érkezik:
 - lamp = 0 esetén a lámpa ki van kapcsolva, tehát be kell majd kapcsolni, ezért a lamp értéke 2 lesz, mely jelzi ezt a főciklusnak, mely bekapcsolás után a változót 1-be állítja
 - lamp = 1 esetén a lámpa be van kapcsolva, tehát ki kell majd kapcsolni, így a lamp értékét most háromra kell állítani, ezzel jelezve a főciklusnak, hogy ki kell kapcsolni azt. A főciklus a végrehajtás után 0-ba állítja a lamp értékét.
 - 'p' működése az 'l'-vel azonos, csak motort kapcsolgatja.
 - 'm' a measurement változót állítja hármas értékre, mely a főciklusnak jelzi, hogy mostantól measurement csomagokat kell küldeni, valamint ha a pumpa vagy a lámpa nincs bekapcsolva, akkor kapcsolja be azokat.
 - 'c' a measurement változót állítja kettes értékre, mely a főciklusnak jelzi, hogy mostantól calibration csomagokat kell küldeni, valamint ha a pumpa vagy a lámpa nincs bekapcsolva, akkor kapcsolja be azokat.

- 'w' a measurement változót állítja egyes értékre, mely a főciklusnak jelzi, hogy mostantól warm up csomagokat kell küldeni, valamint a pumpát ki, a lámpát be kell kapcsolni
- 'o' a measurement változót állítja négyes értékre, mely a főciklusnak jelzi, hogy kell küldeni egy off csomagot, majd utána measurement értékét nullába állítva, a következő ciklusban már No Measurement állapotba fog lépni. Főciklus feladata még a pumpa és a lámpa kikapcsolása is.

Fentiekben bemutatott csomagokon kívül valamennyi mérési állapotban növelődik a status nevű változó, mely ha nagyobb, mint tíz, akkor státusz csomag küldődik és a status értéke nullázódik. Státusz értéke a No Measurement állapotban nem növekszik. Program működésének vázolata a *6.1 ábrán* látható. A program reset után a No Measurement állapotba kerül.

- IO (my_IO): Ebben van definiálva valamennyi használt portokhoz tartozó konstans és művelet. Legfontosabb feladata, hogy definíciók sorozatával könnyen olvasható funkciókkal lehessen vezérelni a motort és a lámpát, valamint a továbbfejlesztést is nagyban segíti. Például a lámpa vezérlésére az előbbiek alapján:

```
#define LAMP PORTC4 //pin definiálása

//láb bemenetként/kimenetként történő inicializálása
#define set_input(portdir,pin) portdir &= ~(1<<pin)
#define set_output(portdir,pin) portdir |= (1<<pin)

//láb kimenet esetén az állapot beállítása
#define output_low(port,pin) port &= ~(1<<pin)//port
#define output_high(port,pin) port |= (1<<pin)

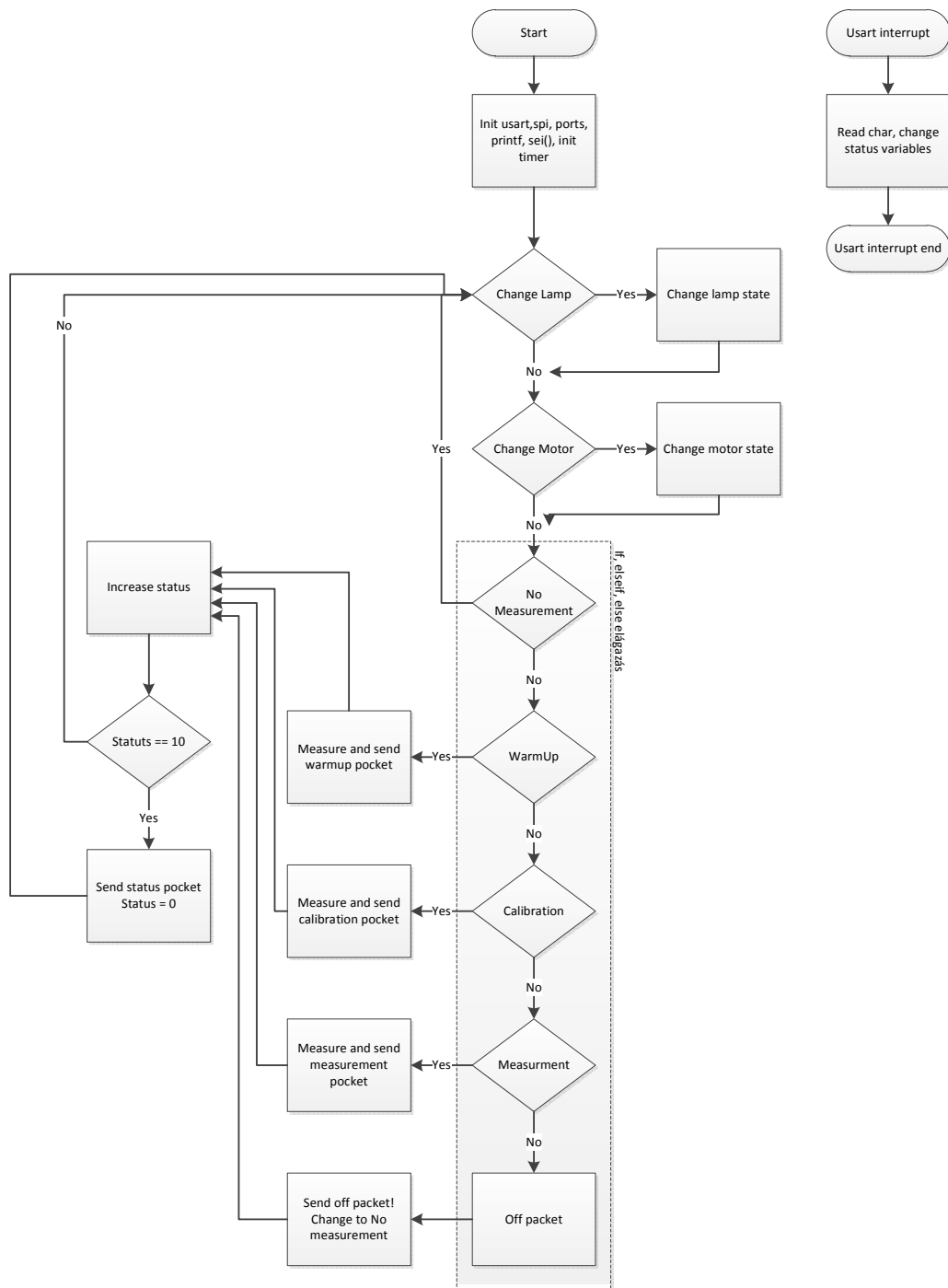
//lámpa init és vezérlő funkciói az előbbiek használatával
#define init_lamp_port set_output(DDRC,LAMP) //init
#define turn_lamp_on output_high(PORTC, LAMP) // turn on
#define turn_lamp_off output_low(PORTC, LAMP) // turn off
```

- Pocket (Pocket.h, Pocket.c): Feladata kommunikációs csomagok definiálása, előállítás, ellenőrző kód generálása és csomagok elküldése. Fontos kiemelni, hogy mivel kevés küldendő információ van, így nyugodtan küldhető olyan formátumban, ami szabad szemmel is könnyen értelmezhető és nincs szükség semmilyen tömörítésre.

Valamennyi csomag kezdő ('<') és lezáró (',>') karakterrel rendelkezik, ha ezek közül bármelyik előfordul adatként, akkor duplázva lesz, ezzel jelezve, hogy az nem határoló karakter. Az adatok előtt azonosító karakter található, mely az olvashatóságot hivatott szolgálni, ez általában az adatra jellemző angol szó első betűje: (F)rame, (P)ressure, (E)xternalCo2, (T)emperature, (R)eference, (C)o2,(L)amp,(M)otor, Check(S)um és (M)easurement Type. Ezután következik az adat és utána egy szóköz karakter, kivéve az ellenőrző kódot, amely után nincsen szóköz. Három, adatokat tartalmazó csomag ellenőrző összeggel is rendelkezik, ami a csomagban található adatok értékeinek összege, és nyolcbites számmá történő konvertálásával áll elő. A státusz csomagok lehetővé teszik a hardver állapotnak ellenőrzését, ami egyben visszajelzés, hogy kézbesítve lettek-e a csomagok. Csomagok (# – a mért értéket, valamint ellenőrző összegét jelöli):

- Méréssel kapcsolatos csomagok:
 - Kalibrációs csomag: <F1 P# E# T# R# C# S#>
 - Mérési eredmények csomagja: <F4 P# T# R# C# S#>
 - Bemelegítési csomag: <F5 T#>
- Státusz csomagok:
 - Állapot csomag: <F7 L# P# M# S#>
 - Mérés vége csomag:<F6>
 - Lámpa be/kikapcsolva csomagok: <F2 L1>, <F2 L0>
 - Motor be/kikapcsolva csomagok:<F3 M1>, <F3 M0>

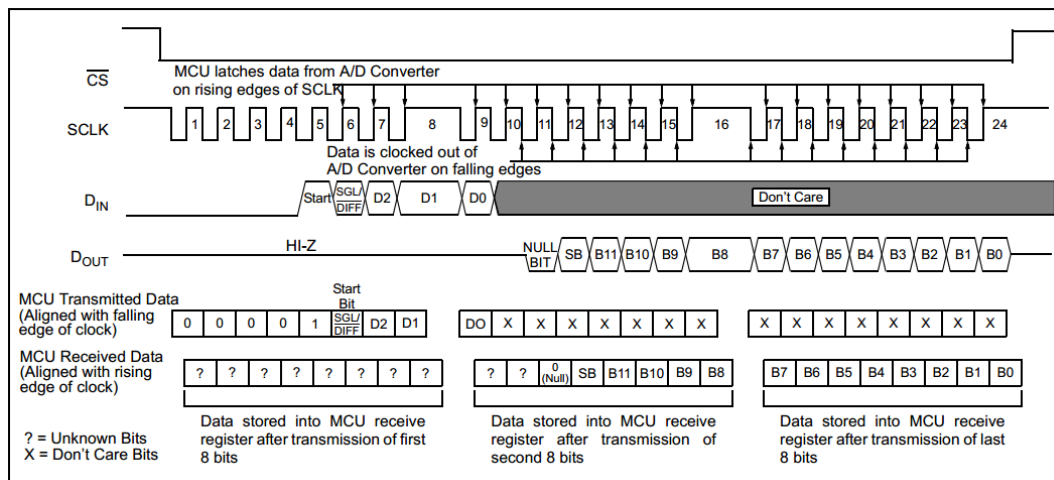
A fentiek mind a hardvertől számítógépnek küldött csomagok. De szükség van az ellenkező irányba közlekedő csomagokra is. Ezek “semmilyen” védelemmel nincsenek ellátva, valamennyi egyetlen egy karakterből áll. Ez érthető is, mert a PC feladata csak az állapotgép valamely állapotának beállítása, vagy a motor és lámpára vonatkozó utasítások. A státuszcsomagok értelmezésével a PC azonosítani tudja, hogy a hardver megkapta-e és feldolgozta-e a kérést. Parancsok már a main résznél be lettek mutatva.



6.1. ábra: Főciklus működésének vázlata

- SPI (spi.c és spi.h) Feladata az SPI periféria vezérlése, amely lehetővé teszi a külső A/D-vel történő kommunikációt. A külső A/D-nek öt bemenetére van rákötve a két thermopile, a termisztor, a nyomá szenzor

és a külső széndioxidmérő. Egy-egy bemenet értékének beolvasása az alábbiak szerint történik:



6.2. ábra SPI kommunikáció menete. Forrás: [27]

1. A/D kiválasztása CS negált lábának földre húzásával
2. Regiszter felső 4 bitjét 0-ba állítani, míg az alsó négy bit: 11 (második egyes jelentése, hogy a bemenetek egyvonalasak) és az olvasni kívánt bement felső két bitje. Ennek hatására megtörténik az adatcsere a mikrokontroller és az A/D között. A beolvasott bájtban közömbös adat van.
3. Regiszter legfelső bitjébe be kell írni az olvasni kívánt bement legalsó bitjét, a többi bit közömbös. Ennek hatására megérkezik az adat felső hat bitje.
4. Regiszterbe újabb beírás hatására megérkezik az adat alsó nyolc bit is. Ezzel megkaptuk a kívánt értéket.
5. A/D CS negált lábát magas szintre kell állítani.

Valamennyi olvasás a fentieknek megfelelően kell végrehajtani.

7 PC-s szoftver

A szoftver C# nyelven íródott, mely a .NET keretrendszer részeként kifejlesztett objektum orientált nyelv. A nyelvet azért választottam, mert eddigiben ezen a nyelven még nem volt lehetőségem programozni. A fejlesztést a szoftver funkcionális részegységekre bontásával kezdtem, majd valamennyi ilyen modult egyesével külön kifejlesztettem, és valamennyihez íródott egy-egy egyszerű valamennyi funkció tesztelését lehetővé tevő konzolos program. Mindenütt törekedtem az általánosításra, ahol csak szükség volt rá, valamint olyan modulokhoz, amelyek megkövetelték, usercontrol is készült, ezzel növelve felhasználhatóságát.

7.1 Szoftver részegységeinek bemutatása

7.1.1 Alap DLL-ek

A szoftver elkezdése előtt pár dll fájl megírásával kezdtem, melyek alapvető funkciókat látnak el, és ezek használata bárhol előfordulhat a programban.

Munkámat egy *DataStorage.dll* nevű dll fájl megírásával kezdtem. Ebben több osztály is található, de csak hármát emelnék ki, melyek mind generikus osztályok. **CriticalVariable** bármely a C# által támogatott típusal létrehozható. Feladata, hogy megakadályozza, hogy a benne tárolt változóhoz egyszerre több szál is hozzáférjen. Másik osztály a **Noteditable**, melynek lényege, hogy a konstruktorában létrehozott értéknek azonnal értéket kell adni, és a későbbiek során annak értéke csak lekérdezhető, változtatásra nincsen lehetőség. Ezzel hasonlóan működik, mint a const, de az értékét a programban első használatkor kell csak beállítani. Ez rendkívül jól használható modulok között, ahol az egyik modul beállíthat egy értéket, míg a többi csak olvashatja azt. Utolsó osztály mely ide tartozik a **Fifo**. Fontos, hogy szálvédett, és egy általános dinamikus méretű FIFO-t (First In First Out adattároló tömb) valósít meg. Ez az osztály sok esetben nyújt egyszerűbb megoldást: például a soros portról beolvasott csomagok a feldolgozás előtt ebben tárolhatóak átmenetileg, valamint egy csúszó átlagolás is átláthatóbban kezelhető használatával.

Másik fontos felad a loggolást lehetővé tevő alkalmazás elkészítése, mely megkönnyíti a debuggolást, valamint kész termék esetén a felhasználónál esetlegesen felmerülő problémák kiderítésére is kiváló. Erre szolgál a *Logger.dll*, mely egy statikus osztályt tartalmaz, amely használja az előző bekezdésben bemutatott osztályokat. Az osztályban definiált statikus függvények segítségével a program bármely részéről tudunk üzenetet és a hozzátartozó prioritási szintet átadni. A prioritási szint a debuggolást nagyban segíti. Alkalmazott szintek:

1. Hiba üzenetek.
2. Warning üzenetek, amelyeket érdemes figyelni, de hibát nem fognak okozni (pl. sikertelen csatlakozás, nem elérhető kiválasztása miatt).
3. Adat üzenetek (pl. mennyi volt egy adott változó értéke).
4. Fontosabb programállapot üzenetek (pl. sikeres csatlakozás).
5. Általános programállapot üzenet (pl. a szoftver belépett egy adott függvénybe).

A loggernek átadott üzenet FIFO-ban tárolódik átmenetileg, amelyből a loggerben futó szál beállított időközönként kivesz az ott található valamennyi üzenetet és fájlba menti. A menteni kívánt legmagasabb prioritási szint megadható, valamint lehetőség van olyan logger indítására is, amely csak adatok formázatlan mentésére szolgál.

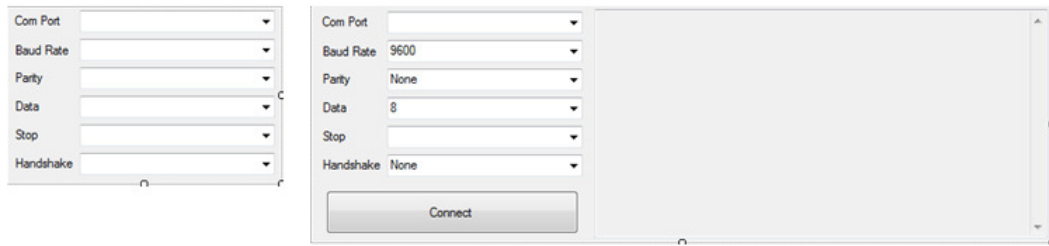
7.2 Szoftver részegységeinek bemutatása

7.2.1 Soros porti kommunikáció

_SerialCommunication modul: Az aláhúzás mindig azt jelöli a modul nevében, hogy az abban található valamennyi osztály általános megoldásokat tartalmaz, így semmilyen feladatspecifikus megoldást tilos elhelyezni benne. Az USB virtuális soros portként látszódik FTDI chip és annak driverének használata esetén. A hardvertől érkező csomagokat ennek a modulnak a feladata fogadni, letárolni feldolgozásig és csomagokra felbontva tárolni.

A modul hat részből áll: három osztályból, két usercontrol-ból és egy interfészből. A három osztály: *SerialCommunicationSettings*, *SerialCommunication*,

SerialCommunicationWithFifo. Az első osztály, a SerialPort osztályból származik, feladata az elérhető soros port beállítások lekérdezése (pl. az elérhető COM – portok nevei). Egyetlen mezője sincs. A vizuális megjelenítéséért a UCSerialCommunicationSettings usercontrol felel. A usercontrolok feladata, olyan formok (Programtervezés során a megjelenítő ablak megfelelője a form) előállítás, melyek könnyen újrafelhasználhatóak, valamint a kód tagolását, átláthatóságát is növelik. UCSerialCommunicationSettings egy általános felületet nyújt a soros port beállítására, valamint egy későbbiekben bemutatott Properties modul segítségével az utolsó beállítás mentésére is képes. A SerialCommuncation osztály szintén a SerialPort osztályból származik. Feladata a soros port kezelése: megnyitása, bezárása, olvasása és írása. A port olvasása a SerialPort DataReceived esemény felüldefiniálásával történik, de ez a függvény is virtuális, hogy következőekben bemutatott leszármaztatás után, a leszármazott osztály azonos nevű függvénye hívódjon meg. A SerialCommunicationWithFifo a SerialCommunication osztályból származik. Ősosztályt kiegészíti kettő darab FIFOval és egy csomagok darabolását végző függvénnyel. A soros portra érkező új adat a port FIFOjában tárolódik. Ha a csomagdarabolás engedélyezve lett a konstruktorban, akkor az új adat lementése után egy újabb esemény generálódik, melynek hatására a daraboló függvény kiveszi az addig beérkezett összes adatot a torolóból és a beállított kezdő és végkaraktereknek megfelelően csomagokra bontja az adatfolyamot és azt a csomagoknak fenntartott FIFOba teszi. A csomagokat a későbbiekben bemutatott feldolgozó modul veszi ki, mely az osztály egyetlen függvényéhez fér hozzá az IReadCommunicationFifo interfészen keresztül, mellyel vagy a csomag vagy a soros port FIFOját képes olvasni. UCSerialCommunication usercontrol használja a már bemutatott UCSerialCommunicationSetting-et. Ennek használatával megadható a soros port beállításai, majd a kiválasztott porthoz történő csatlakozás után, a beérkező csomagok a kontrolon található textbox-ban jelennek meg. Ha textbox rendelkezik a fókusszal, akkor a begépeltek karakterek továbbítódnak a soros porton keresztül.



7.1. ábra: UCSerialCommunicationSettings és UCSerialCommunication usercontrok

7.2.2 _Dataprocess és Dataprocess modul

A SerialCommunication által tárolt adatok további feldolgozást igényelnek. Ezt a feladatot látja el a Dataprocess modul. A _Dataprocess konstruktorában megkapja a SerialCommunication modul interfészét, melyen keresztül a beolvasott csomagokat ki tudja olvasni. Konstruktoron kívül még három függvényt tartalmaz, egy szálindítást és egy szálleállítására alkalmas függvényt, valamint a virtuális Dataprocess függvényt, melyet a szál használ futása során. Ez a függvény a _Dataprocess-ben üres, hogy az általános megvalósításnak eleget tegyen. A Dataprocessben az ebből leszármaztatott osztály végzi a tényleges feldolgozást az azonos nevű függvényében: beolvas egy újabb csomagot, megnézi a csomag azonosítóját és annak megfelelő feldolgozó függvénynek továbbítja, mely kinyeri az adatokat, ellenőrzőösszeg esetén leellenőrzi azok helyességét, ha hibás, akkor eldobja, egyéb esetben a kapott üzenetben szereplő adatokat továbbítja eseményként vagy hardver állapotát jelző változókon változtat.

7.2.3 _Properties modul

Feladata felhasználói beállítások mentése a registry-be és visszatöltése onnan. Működése hasonló a Javában használt Properties-zel: Subkey beállításával meg kell adni a mentés helyét a registry-n belül. Valamennyi lekérdezés függvénye kétparaméterű, első paraméter a lekérdezni kívánt változó nevét, második paraméter egy alapértelmezett értéket tartalmaz. Az alapértelmezett érték használatával elkerülhető az első lekérdezés kezelésének problémája. Ha a lekérdezett változó létezik a registry-ben, akkor visszaadja a függvény annak értékét, ha viszont nincsen ilyen, akkor rögzíti a változó nevet és alapértelmezett értéket, valamint azzal is tér vissza.

7.2.4 Calculation osztály

Itt található valamennyi szenzor értékének meghatározására szolgáló számítás.

ADC osztály egy-egy A/D érték feszültség értékévé történő konvertálásában segít. Három értékre van szüksége a feszültség érték meghatározásához: az átalakító bitszámára, referencifeszültségére és a mért értékre. Első két érték a konstruktorban adható meg, majd a `ConvertAdcValue(double value)` hívogatásával kapható meg a keresett értékek.

Ezzel csak a feszültség szintet kaptuk meg, ezt még tovább kell alakítani, hogy a ténylegesen keresett, adott szenzorra jellemző számot megkapjuk. Erre szolgálnak a `CO2`, `ExtCo2`, `Pressure` és `Thermistor` osztályok. Valamennyi osztály az adott szenzorra jellemző konstans adatot konstans paraméterként tartalmaz. Mindegyik osztálynak egyetlen számolást végző függvénye van, amelyek bemenetként a mért ADC értéket/értékeket kell megadni, ezeket aztán a függvény az ADC osztály segítségével átváltja feszültséggé, majd elvégzi rajtuk az adott szenzorra vonatkozó számolást.

Fontos megemlíteni, hogy az egyes szenzorokra jellemző változó értékeket: a nyomásszenzor nullpontját, a termisztor *RO* és *TO* értékét valamint a `CO2` két kalibrációs pontjához tartozó nyomás, hőmérséklet és széndioxid szint értékek beállíthatósága, megváltoztathatósága mérési eredmény pontosabb meghatározását teszi lehetővé. Ezeket az értékeknek is van megfelelő `_Properties` modul segítségével tárolt értéke, ezek az adott szenzorra jellemző kalibrációs értékek. Ezekhez az értékekhez a `CalibrationData` osztályon keresztül férnek hozzá a számoló/kalibráló osztályok.

7.2.5 Charts

A .NET rendelkezik egy `Charting` nevű kontrollal, mely nem része a telepített keretrendszernek, de letölthető és egyetlen `dll` betöltésével használható. Ez egy rendkívül sok funkcióval rendelkező modul, melyek között könnyen elvesz a felhasználó. Valamennyi beállítás elérhető a tulajdonság ablakokon keresztül, de ezek ily módon történő beállítása elveszi a dinamikusan változtatható megjelenítő használatának lehetőségét.

A `Chart` egy ablakában több görbe terület is definiálható, ezeket `Chartarea`-nak nevezi. Ezek száma bármennyi lehet, de csak azokat jeleníti meg, amelyekhez pontok is

tartoznak. Egy görbének a Series felel meg. Csak azok a Series-ek jelennek meg, amelyek hozzá lettek adva valamely Chartarea-hoz. Fontos még, hogy mind a Chartarea-ra, mind a Series-re a létrehozásukkor megadott névvel tudunk hivatkozni. A Series-ben tárolódnak a pontok, ahány pontot tartalmaz, annyit is fog megjeleníteni. Tehát, ha görbének csak egy adott részét szeretnénk látni, akkor pontokat kell kitörölnünk, így azok el is vesznek. Ezért lett a Chart pár új osztállyal kiegészítve.

A Chart dinamikus használatát egy ChartWrapperClass elkészítésével vált lehetővé. Ennek működését három további osztály segíti: PointBackUp, PointBackUpHandle, SaveDataToExcel. Az utolsó működése egyértelmű, a görbéken megjelenített pontok exportálhatóak ki Excelbe. A PointBackUp-ból mindig létrejön egy példány, amikor létrejön egy Series, és rá a Series nevével lehet hivatkozni. Feladata valamennyi pont tárolása, melyet az adott Serieshez hozzáadunk, valamint annak tárolása, hogy ezek közül éppen melyik tartomány jelenítődik meg. Ez lehetővé teszi, hogy egy-egy tartomány jelenítsünk csak meg és mégse veszítünk el pontokat. A PointBackUp-ok listáját kezeli a PointBackUpHandle. Mindkettőt a ChartWrapperClass kezeli.

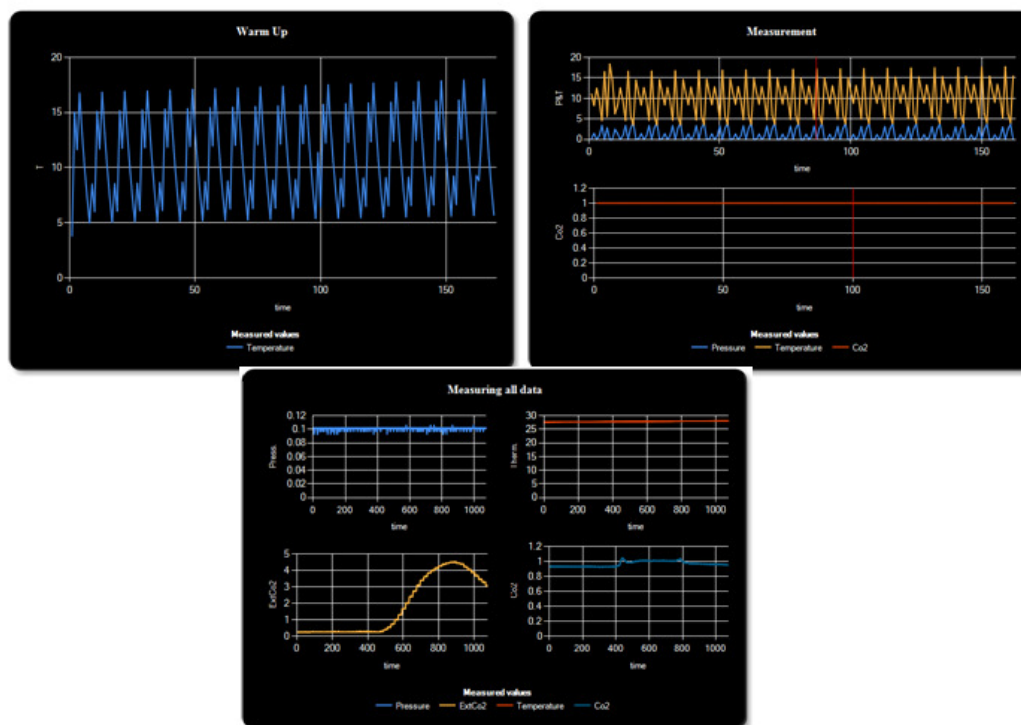
A ChartWrapperClass pontos bemutatása nagyon hosszadalmas feladat lenne, így csak a fontosabb funkciókat sorolom fel:

- ChartArea hozzáadása a Chart-hoz
- Series létrehozás és ChartArea-hoz adása
- Series átrakása egy másik ChartArea-ba
- ChartArea elrejtése/megjelenítése
- Chart-hoz tartozó legend megjelenítése
- Chart név megjelenítése
- Pontok értékeinek megjelenítése a görbék felett
- Görbék zoomolásának engedélyezése
- Görbék megjelenítése görbeként/pontok halmazaként
- Egyetlen kiválasztott Chartarea megjelenítése
- Koordinátarendszerek felíratainak beállítása

- Chart nyomtatása
- Chart képbe mentése
- Pontok exportálása Excelbe.

CO2_Chart a modul feladat specifikus usercontrol. Chartot a ChartWrapperClasson keresztül éri el. Feladata a chartok dinamikus megjelenítése. Három megjelenítési módot támogat: kalibrációs, mérési és bemelegítő. Mindhárom esetben más-más görbék megjelenítésére van szükség. A fentiekben felsorolt funkciókon kívül a következő funkciók adódnak a modulhoz a usercontrol használatával:

- Beállítható a görbéken látni kívánt pontok száma
- A görbék megjelenítése elindítható/leállítható
- Dupla kattintással csak kiválasztott görbeterület látszik
- Egy-egy pontra kattintva annak értéke lekérdezhető
- Valamint, ha görbénk csak egy szakasza látszódik, akkor a jobbra-balra eltolható. Ennek kalibrációnál van főleg lényege.



7.2. ábra: Co2_chart usercontrol három állapotának képe

7.2.6 Szimulátor

A szoftver fejlesztése során gyakran van szükség a hardver felesleges használatára, mely sokszor kényelmetlen, hisz egy fejlesztési példány folyamatos jelenlétére van szükség, mely kezdetleges állapota miatt sérülékeny, nehezen mozgatható. Ugyancsak sokszor előfordul, hogy nem lehet megállapítani, hogy a szoftver vagy a hardver generál-e egy adott hibát. Mindkét esetre jó megoldás a hardver viselkedését szimulálni képes független program. Használta ugyancsak hasznos, ha egy valós mérés közben, adódik valamilyen ismeretlen hiba, mivel szimulátor használatával könnyen ismételhetővé válik, valamint ellenőrizhető, hogy sikerült-e a javítás.

A Loggernél volt szó róla, hogy képes csak a mérési eredmények fájlba mentésére is. Ez a funkció, az előző bekezdésben említett hiba esetén rendkívül hasznos. Tesztelési ciklus alatt valamennyi mérés adat rögzítése, sokban segíti a szoftverfejlesztést és hardvertesztelést.

Mindezek miatt készült el a szimulátor, melynek szoftverének központi eleme egy az egyben a mikrokontrolleren futó főciklus kódja. A megszakításokat események, az időzítőket, pedig azoknak közel azonosan alkalmazható szoftveres időzítők helyettesítik. A C# a C-ből származik, így C kódok felhasználhatóak, egyetlen fontos különbség a típusváltozókban van, melyek megválasztására érdemes odafigyelni.

A PC-s szoftver csak elindítani és leállítani képes a szimulátort, egyéb műveletek elvégzésére csak a soros porton keresztül küldött üzenetekkel tudja utasítani. A PC-s szoftver és a szimulátor kapcsolatát lehetővé tevő virtuális sorsportokat generáló szoftver nem része a diplomatervnek.

A szimulátor a mérési adatokat korábbi mérések eredményeiből nyeri, valamint, ha ezek nem elérhetőek, akkor véletlenszám-generátort használtam azok előállítására.

7.2.7 Modulok összekapcsolása: DataCenter és Controller osztályok

DataCenter osztályaiban, Constants, HardwareStates, Settings, States, található valamennyi modul azon állapotai, amelyek fontosak lehetnek a többi modul számára. Ugyanitt tárolódik a program különböző részein használt nem osztályspecifikus

konstansai, definíciói és felsorolt állandók (enum). Fontos megjegyezni, hogy valamennyi itt elhelyezkedő változó szálvédett.

A Constans osztály tartalmazza a konstansokat, ilyenek a felhasználói beállításokhoz tartozó regisztry bejegyzések nevei. Ugyancsak itt tárolnak a hardvernek küldött vezérlő parancsok karaktereinek definíciói is, így például a soros portra nem egy 'l' karaktert kell kiküldeni, hanem egy Constants.lighCommand-ot, amellyel a kód is olvashatóbbá és változtathatóvá válik. Ebben az osztályban találhatóak a delegate változók definíciói és a közös felsorolt állandók.

HardwareStates a hardver állapotait tárolja. Ezeket a soros porton érkező csomagokból kiolvasott értékek alapján állítja be a csomagfeldolgozó. Egy-egy érték megváltoztatásának hatására frissül a központi ablakban lévő, hardver állapotait megjelenítő UCHardwareStatus usercontrol is.

Settings jelen állapotban csak a soros porton beállított adatok mentésére és visszatöltésére szolgál. A beállítást lehetővé tevő settings ablakban a beállított értékek bezáráskor ide és a registry-be íródnak, valamint majd innen olvasódnak vissza. Az osztály létrehozásakor feltöltődik a legutolsónak használt beállítások alapján.

States osztályban csak két változó található, melyek tárolják, hogy szimulátor fut-e, valamint, hogy a soros port meg lett-e nyitva már.

Contoller osztályaiban, CalibrationControl, HardwareControl, NewDataControl, függvények találhatóak, melyek működése hasonlít egy-egy eseményre.

CalibrationControl a kalibrációkor keletkezik, bemenetként megkapja a mért kalibrációs és saját széndioxidmérő, nyomásmérő és hőmérő adatait, majd ezek alapján kiszámolja a saját széndioxidmérő egyes értékeihez tartozó tényleges széndioxid értékeket. Ennek kiszámolása során csúszó átlagolást, hőmérséklet- és nyomáskompensációval is számol, melyek elkérhetőek az osztálytól ellenőrzés végett.

HardwareControl feladata a hardverrel történő kommunikáció. Itt található a csatkozás felépítéséhez és annak bontásához szükséges műveletek, valamint valamennyi vezérlő utasítás függvénye, melynek bármelyik meghívása esetén a program leellenőrzi, hogy létrejött-e már a csatlakozás, és ha kell, elindítja a megfelelő feldolgozó, olvasó szálakat is.

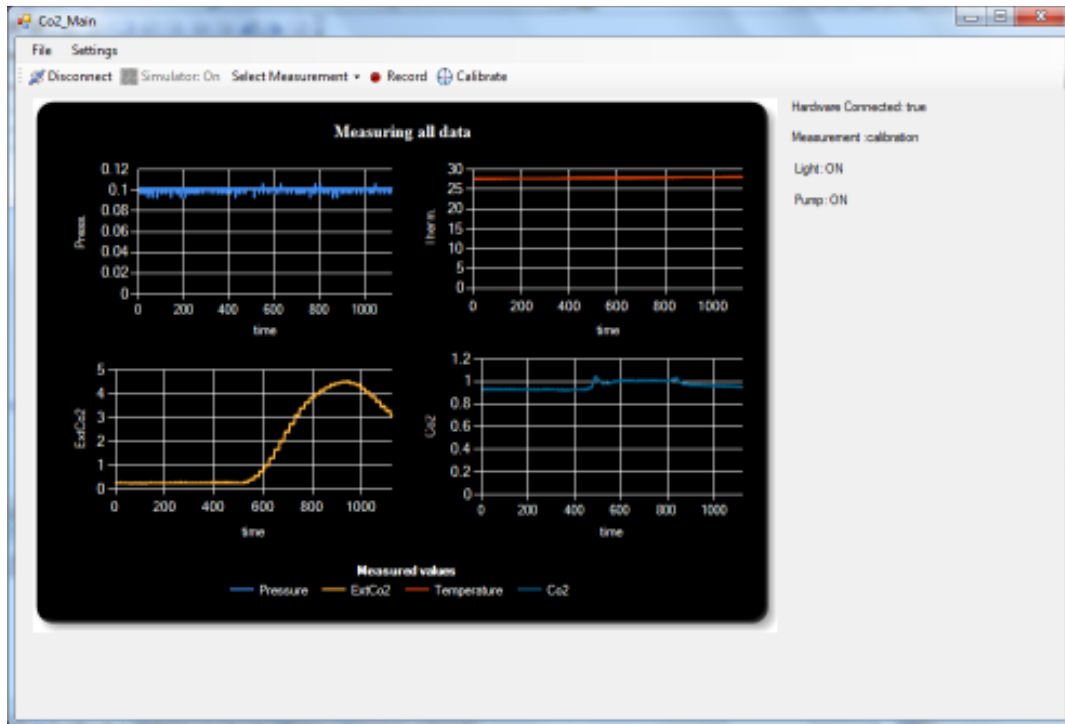
NewDataControl-t a csomagfeldolgozó használja, egy adatcsomag feldolgozása után, meghívja a megfelelő függvényét, és az végrehajtja a pont(ok) mentését, valamint átadja azt/azokat a Chart modulnak.

7.2.8 Központi megjelenítő modul

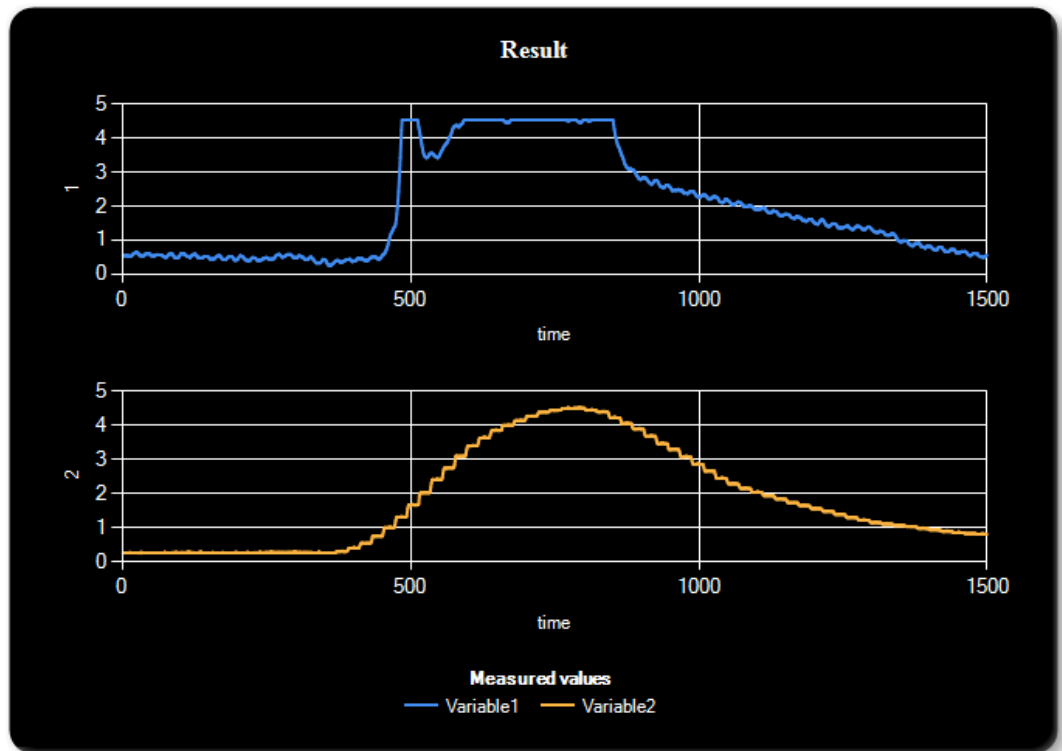
Itt található a központi megjelenítés, mely a felhasználói események kezelésére szolgál. Adatokat nem tárol, így teljesen megvalósítja a Document-View felépítésben előírtakat.

Két form-ot tartalmaz, ComPortSettings és CO2_Main-t. Az elsőt nem részletezném, mert feladata neve alapján egyértelmű, valamint a korábbiakban bemutatott USSerialCommunicationSettings-t használja.

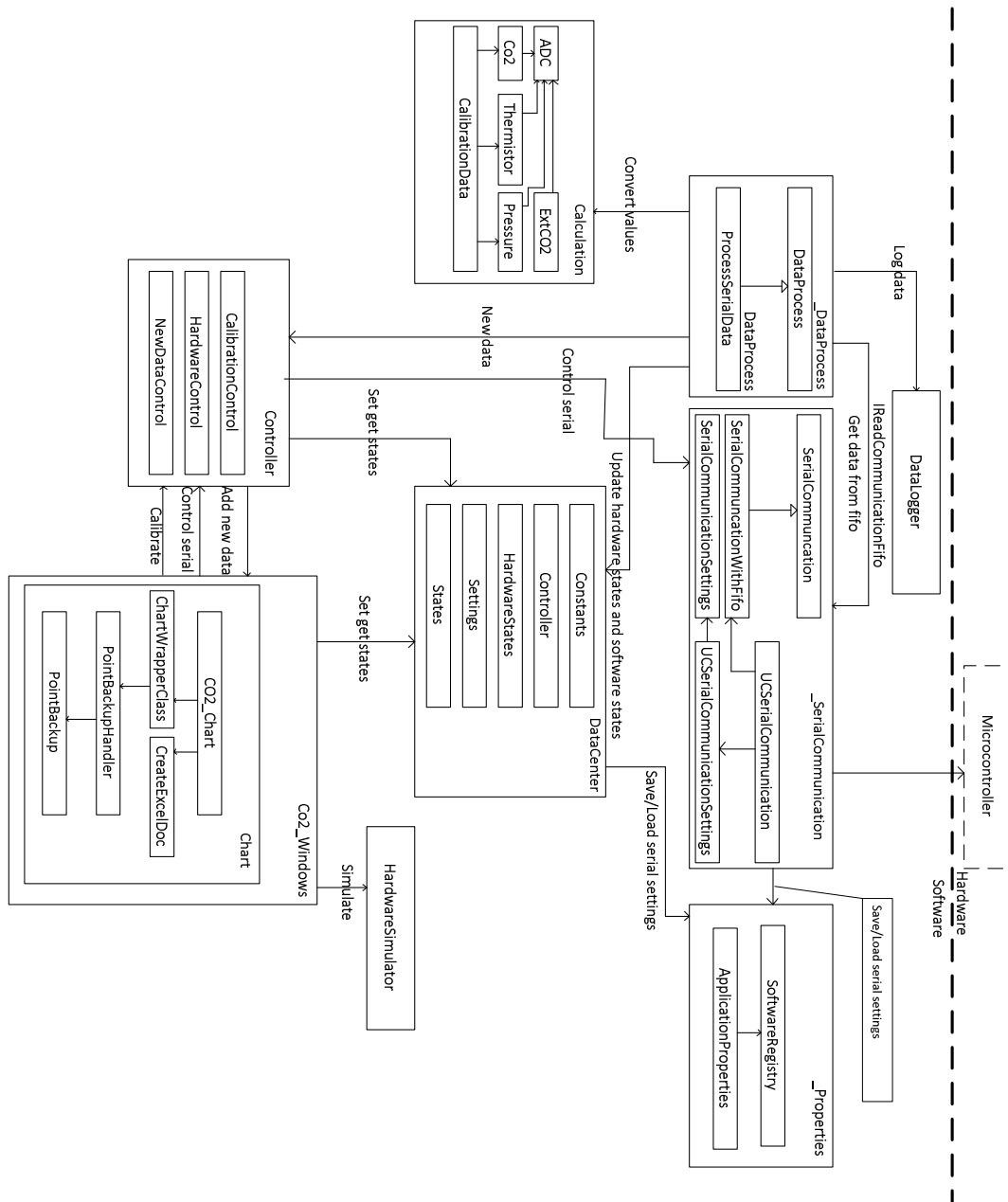
A főablak képe a 7.3 ábrán látható. Ha nincs kéznél hardverünk, akkor a szimulátor elindításával kell kezdenünk a „Simulator: off” gomb lenyomásával, későbbiekben ugyanezzel állítható majd le. Megint kiemelném, hogy külső virtuális soros port generáló szoftverre van szükség, hogy használni lehessen. Ezek után a szimulátoros és hardveres felhasználás megegyezik. A „Connect” gomb kiválasztásával lehet a csatlakozást létrehozni. Majd „Select Measurement”-ből ki kell választani az elvégezni kívánt mérést, melynek hatására a kiválasztott mérésnek megfelelő megjelenítőre vált a Chart. A pontok rögzítésének elindítása, leállítása a „Record”/„Stop” gomb kiválasztásával történik. Leállított állapotban a Chart már bemutatott funkció érhetőek el. Kalibrációs mérés leállítása után a Calibrate gomb aktívvá válik. Ennek megnyomása előtt megadható a pontok száma, melyek alapján a kalibrációt számoltatni kívánjuk, valamint azok egymáshoz képest eltolhatóak. Fontos megjegyezni, hogy a hőmérséklet, nyomás és az épített széndioxid mérési eredményei összetartoznak, így azok együtt mozognak eltolás esetén. A Calibrate gomb megnyomásának hatására a kalibráció végrehajtódik és az eredmény megjelenik a Chart-ban. Erre látható péda a 7.4 ábrán. A szoftver modulok és osztályok kapcsolata a 7.5 ábrán látható.



7.3. ábra: Főablak



7.4. ábra: A legelső mérés alapján kapott kalibráció ábra



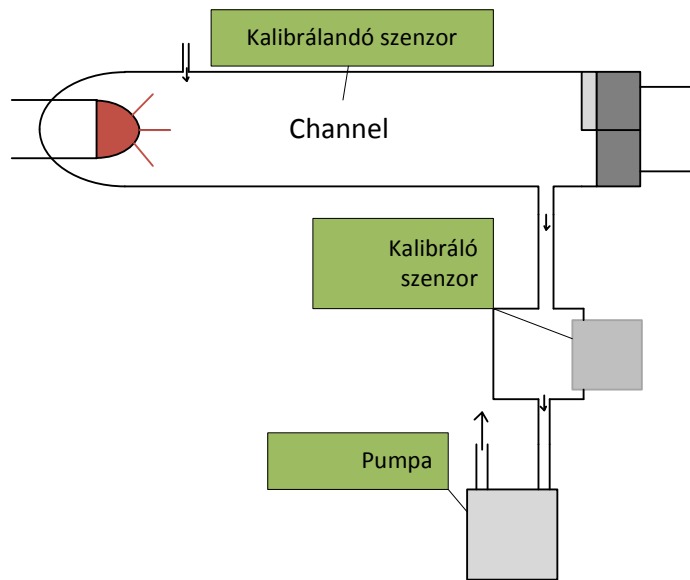
7.5. ábra: Program osztályainak, moduljainak kapcsolata (nyilak a használtól a használt felé mutatnak)

8 Kalibráció és szenzor tesztelése

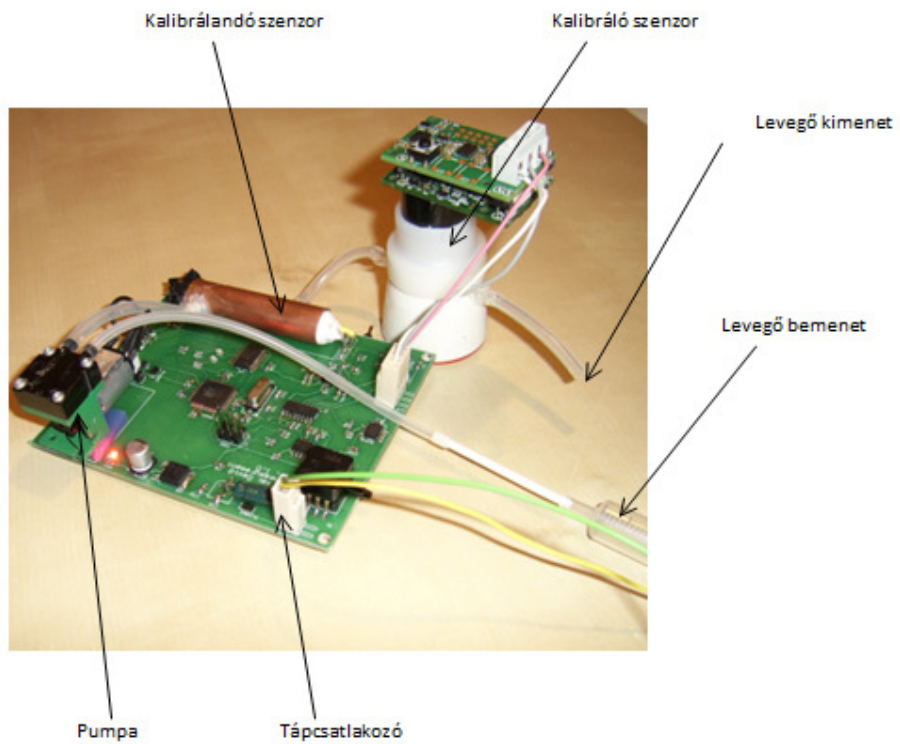
A kalibrációs eljárások a 2.3.3. fejezetben részletesen be lettek mutatva. Kalibrációs gáz használatával lehetett volna a legjobb eredményt elérni, de ez elég költséges, egyetlen palack értéke legolcsóbb esetben is ötven – hatvan ezer forint, így ennek alkalmazásáról lemondtam. Ezért a másik kettő kalibrációs módszereket próbáltam ki.

8.1 Külső kalibrációs széndioxid használata

Ennek a módszernek a lényege, hogy a kalibrálni kívánt és a kalibrációs eszközt összekapcsolom egy csővel és rajtuk keresztül az áramlást a pumpával biztosítom. Bekapcsolás után a rendszert hagyni kell szabadon járni, hogy szenzorok bemelegedjenek, és a külső levegővel átszellőződjön a rendszert. Bizonyos idő elteltével mindkét szenzor közel állandó értékre fog beállni, ezzel meg is van ez első kalibrációs pont. Ezután kilélegzett levegőt lassan, nagy nyomás elkerülendő a bemeneti csőbe belevezetve mindkét szenzor felvesz egy maximális értéket, majd ahogy szépen lassan eltávozik a kilélegzett levegő, akkor az érték csökkenni kezd. Az így kapott maximális érték a második kalibrációs pont. A két pont között a szenzor értéke közel lineárisnak vehető.



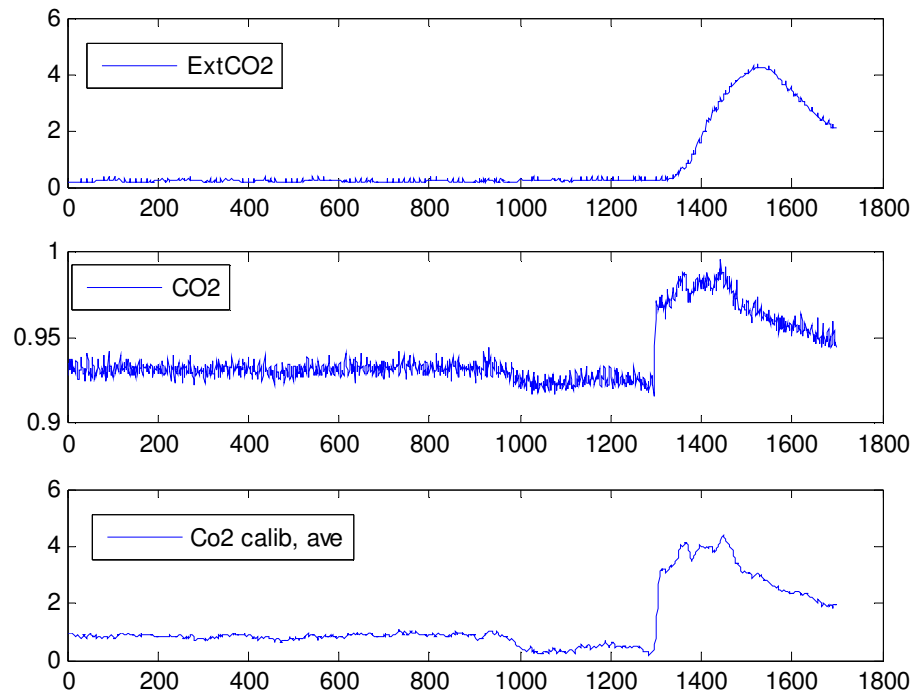
8.1. ábra: Kalibrációs elrendezés



8.2. ábra: Kalibrációs elrendezés

A fentiekben leírt mérést a 7 fejezetben bemutatott szoftverrel végeztem, és a nyers mérési adatok Excelbe exportáltam csv (coma separated values) formátumban, hogy a tényleges számolást független program végezze el.

A csv fájlból Matlabbal betöltöttem a két széndioxid szenzor értékeit egy-egy vektorba (ExtCO2 és Co2). Ezek után egy tízes csúszó átlagolást alkalmaztam a CO2 vektoron, majd megkerestem mindkét vektorban a minimumokat és maximumokat és ezeket összerendeltem. Ezek alapján egy lineáris egyenlet írható fel, melynek használatával a kalibrálandó széndioxid értékei kiszámolhatóak. A három vektor eredménye látható a 8.3 ábrán. A legfelső képen látható a külső szenzor mérési értékei, középen az épített szenzor nyers adatai, míg alsó ábrán a szűrés és kalibrálás utáni értékei láthatóak.

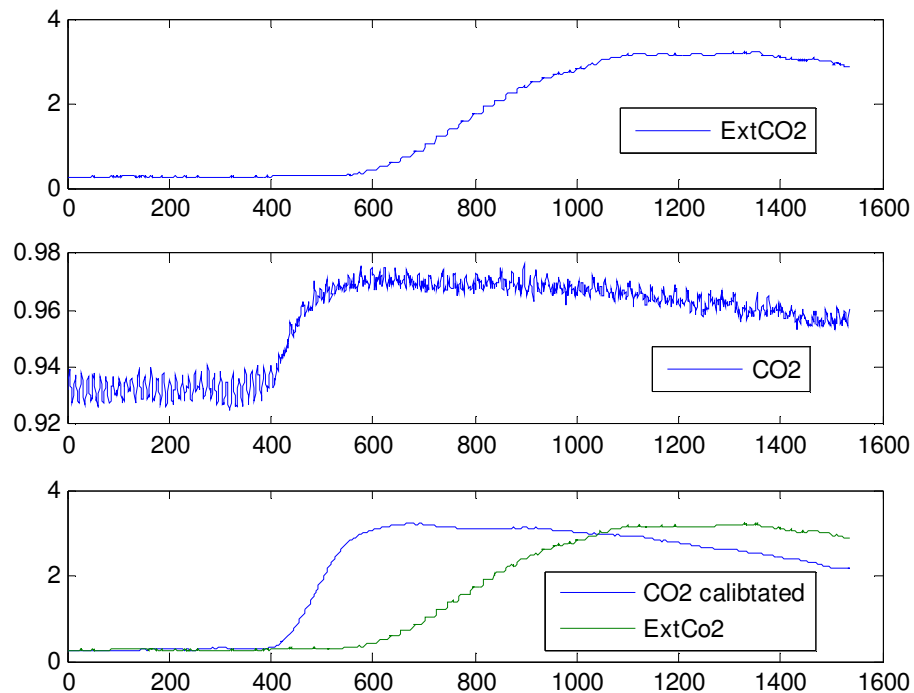


8.3. ábra: Kalibrálás, nyomásváltozás probléma

Eredményekből levonható, hogy szenzorunk sokkal gyorsabb, mint a kalibráló szenzor, valamint az cső késleltetése is megfigyelhető. Látható, hogy a kalibrálás nem teljesen sikeres, ez az 1000-1300 időpontok között jellesés eredménye (lásd 8.3. középső ábra). Ez az esés annak köszönhető, hogy a kifújt levegőt egy nagyméretű fecskendőben lett összegyűjtve, majd elé egy kisméretű fecskendő lett elhelyezve, és az

lett csatlakoztatva a szenzorokhoz vezető csőhöz. Amikor a nagy fecskendő összenyomása megkezdődött, akkor nagyban megnőtt a nyomás, és ez a mérési értékekben esést eredményezett. Ennek pontos magyarázata a 2.3.2.4. részben leírtak alapján érthető. A fejlesztés további fázisában ezen probléma valószínűleg korrigálható, ugyanis a panel rendelkezik egy nyomásmérő szenzorral is, amelynek a segítségével a változó nyomás hatása kiküszöbölhető.

Ezek után változtattam a mérésen, elhagytam a fecskendőt és közvetlen a száj elől lett a levegőt mintavételezve, cső végének száj elé helyezésével. Az így kapott eredmények 8.4 ábrán láthatóak. Sikerült a nyomásváltozás következtében bekövetkező esés kiküszöbölésére, valamint látható, hogy szenzorunk alakja szépen követi a kalibráló szenzort, csak sokkal gyorsabban folynak le a változások.



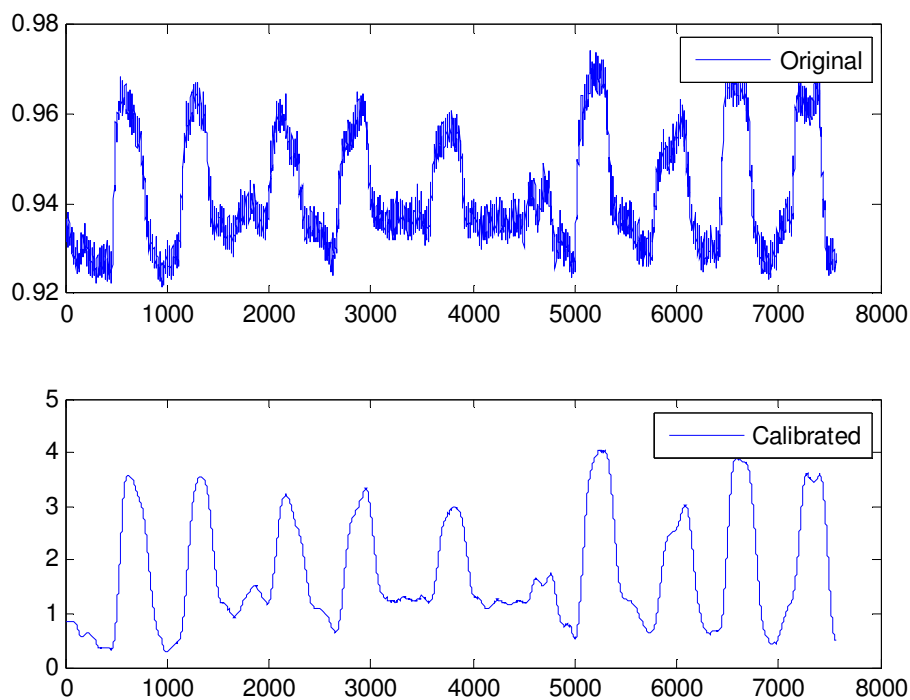
8.4. ábra: Javított mérés

Ekkor logikus a következő lépés, hogy az itt kiszámolt értékeket használjuk egy újabb mérés során és megnézzük az eltérést. Ezeket a méréseket elvégeztem és közel jó értékeket kaptam, de sok egymás követő mérés során kiderült, hogy a kalibrációs szenzor nem jó referenciának, mert felső értéke jócskán változik. A kielégített

levegőben sok mérés során gyakran előfordult, hogy 6% körüli értéket mért, mely nem elfogadható. Alsó értéke közel azonos értékre állt be.

8.2 Kalibráció környezeti levegő és kifújt levegő használatával

A mérés menete, hogy szenzoromba vezető cső előtt lélegzek ki, majd elveszem azt a számtól a levegő beszívás alatt, amennyire csak tudom. Ezek után megvárom, míg kiürül a csatorna, majd újra befújok és ezt ismétlem. Az így kapott görbe minimum és maximum értékét becslem 0,4 és 4,04% szinttel, mely egy elfogadott normál érték. Ekkor a 8.5 ábrán látható eredményt kaptam.



8.5 ábra: Kalibráció kilélegzett levegővel

Az eddigi mérési eredmények alapján látható, hogy az eszköz kalibrálása nem triviális feladat, és jelen dolgozatban nem is közlök teljes mértékben tökéletes kalibrálási megoldásokat. Habár a bemutatott módszerek jó közelítéssel megfelelő nagyságrendű mérési eredmények előállítását teszik lehetővé, de a kalibrálási módszer kidolgozása még további fejlesztési és kutatómunkát igényel, és akár új eszközök beszerzése sem biztos, hogy elkerülhető. Mindazonáltal az általam elkészített rendszer

alkalmas a nyers mérési eredmények gyűjtésére, valamint azok bármilyen transzformációjára, így a megfelelő kalibrációs módszer kidolgozását követően a széndioxid szintjének mérése egyszerűen megoldható.

9 Összegzés, további fejlesztési lehetőségek

A dolgozatomban összegyűjtöttem a szükséges tudást egy szén-dioxid szenzor fejlesztéséhez, majd bemutattam annak hardvertervezési lépéseit, ezt követően megépítettem a hardvert, elkészítettem az azt működtető firmware-t és megírtam egy tesztelő szoftvert.

A megtervezett rendszer teljes mértékben kielégíti egy elsőszeriás prototípus követelményeit. Egyértelműen belátható a nyolcas fejezet alapján, hogy az eszköz szépen követi a széndioxid változásait és sebessége is megfelelő, de rengeteg további lépésre van szüksége egy végleges termék elkészültéig.

Legfontosabb feladat a kalibráció megoldása. Ki kell találni egy olyan megoldást, amely költséghatékony és átmenetileg jól használható a tesztelés során, mivel kalibrációs gáz megvétele csak a végleges fázisban érdemes, ugyanis az gyorsan elveszíti pontosságát.

A későbbiekben a firmware feladata lesz valamennyi szenzorparaméter kiszámolása, amihez szükség lesz a kalibrációs értékek tárolására is. Feladatai ugyancsak bővülni fognak egy offszet korrekcióval is, mely a lámpa fényerősségében történő változások figyelésével történik.

A szenzor mechanikáján is változtatni kell. A ki- és bevezetéseknek légmentesebben kell záródniuk és rögzítésükre is jobb megoldást kell találni. Valamint a nyomtatott áramkörön érdemes lesz a nyomásszenzort áthelyezni a mintavételi csatornához, mert így korrekcióra és ki-be légzés idejének detektálására is alkalmas lesz.

A PC-s szoftverben a felhasznált diagramok kirajzolásában olykor fellépő késleltetéseken kell javítani, valamint a kiforrott kalibráció eljárás kidolgozása után azt implementálni kell. Emellett szükség van még egy adatbázis elkészítésére is, mely az adatok mentését, visszatöltését teszi lehetővé.

Összességében az elkészült elsőszeriás prototípus túlszárnyalja az elvárt eredményeket, és az eredmények arra engednek következtetni, hogy további fejlesztések után kész termék is lehet majd belőle.

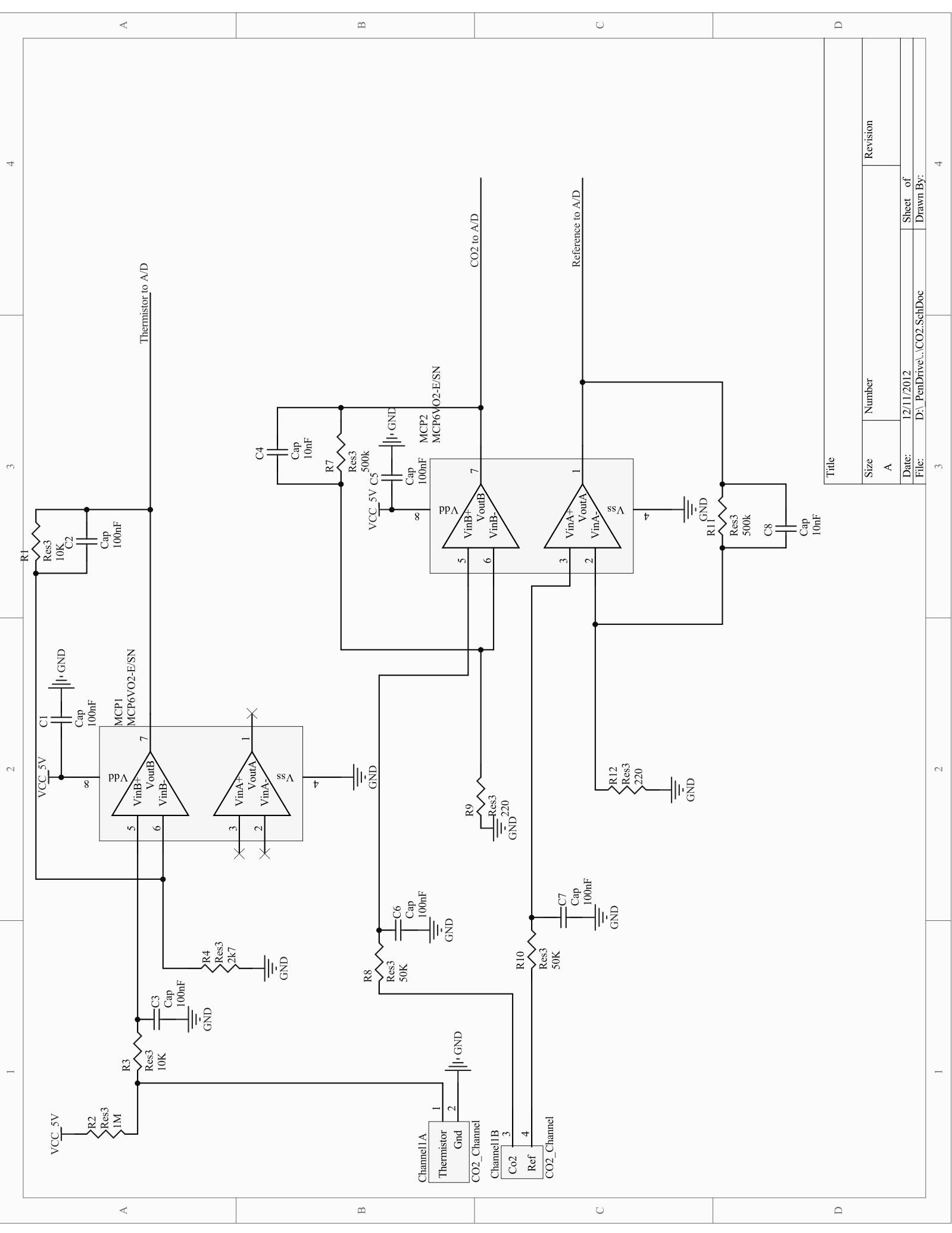
Irodalomjegyzék

- [1] Gyógyszerészeti és Egészségügyi Minőség- és Szervezetfejlesztési Intézet Informatikai és Rendszerelemzési Főigazgatóság: *OECD egészségügyi adatok 2010*,
<http://www.eski.hu> (2012.november)
- [2] Cellscope, Inc
cellscope.com (2012.szeptember)
- [3] Air Quality in Europe: Definition
http://www.airqualitynow.eu/pollution_home.php (2012.május)
- [4] National Heart Lung and Blood Institute: *What Is Asthma?*
<http://www.nhlbi.nih.gov/health/health-topics/topics/asthma/>(2012. november)
- [5] World Health Organization: Chronic respiratory diseases
<http://www.who.int/respiratory/copd/en/index.html> (2012.november)
- [6] National Cancer Institute: *Lung Cancer*
<http://www.cancer.gov/cancertopics/types/lung> (2011.december)
- [7] World Health Organization: *Cardiovascular disease*
http://www.who.int/cardiovascular_diseases/en/ (2011.november)
- [8] Élet és Tudomány: *Szív és érrendszeri betegségek-Életmentő megelőzés*
http://www.eletestudomany.hu/content/aktualis_20081114_sziv_es_errendszeri_betegsegek_Eletmento_megelozes (2012.november)
- [9] SZTE Klebelsberg könyvtár: *A légzés mechanizmusa*
http://xenon.bibl.u-szeged.hu/~vidaa/holi/02/elettan/tetelek/25_legzes.pdf
(2012.november)
- [10] Eastern Kentucky University: *Human Physiology Respiration*
<http://people.eku.edu/ritchison/301notes6.htm> (2012.november)
- [11] Thormed Kft.: *Thorsoft szoftver help*, 2012
- [12] Wikipedia: *Carbon dioxide sensor*
http://en.wikipedia.org/wiki/Carbon_dioxide_sensor (2011.november)
- [13] Billes Ferenc: *Rezgési spektroszkópia*, BME: Fizikai és Anyagtudományi tanszék, 3. , javított és kiegészített kiadás, Budapest 2010
- [14] Veszprémi Egyetem Analitikai Kémiai Tanszék: *Az infravörös és raman spektroszkópia alapjai*
<http://www.muszeroldal.hu/measurenotes/mink.pdf> (2011.december)
- [15] KFKI: 6.2.6. Optikai spektroszkópia
<http://www.kfki.hu/elftvakuum/pub/wplazma/plaz626.htm> (2011. december)

- [16] Szegedi Tudományegyetem Természettudományi és Informatikai Kar:
Tömegspektroszkópia jegyzet
http://www2.sci.u-szeged.hu/inorg/AnalKemEa_MS.pdf (2012.november)
- [17] Absolute astronomy: Total organic carbon
http://www.absoluteastronomy.com/topics/Total_organic_carbon
(2011.december)
- [18] United States Patent: *NDIR instrument*, Patent Number:6,1147,00; 2000.
szeptember 05.
- [19] Parkin Elmer: *Infrared Sensing*
http://www.perkinelmer.co.uk/CMSResources/Images/44-4345CAT_SensorsAndEmittersInfraredSensing.pdf (2012.szeptember)
- [20] E2V Technologies: *Infrared gas absorption spectra*, 2007. május
- [21] United States Patent: *Self-Calibration of an NDIR Gas Sensor*, Patent Number:
5,347,474; 1994 szeptember 13.
- [22] United States Patent: *Cardiopulmonary exercise testing apparatus and method*,
Patent Number: 6,174,289, 2001. január 16.
- [23] United States Patent: *Calibration method for NDIR equipment and calibration
apparatus*, Patent Number: 5.850.354, 1998. december 15.
- [24] Dr. Boros Mihály, *Gyakorlati orvosi alapismeretek*, Egyetemi tankönyv, Szegedi
Tudományegyetem Általános Orvostudományi Kar Sebészeti Műtéttani Intézet,
Szeged, 2007
- [25] Magyar Máltai Szeretetszolgálat: *Kapnográfia*
http://mentoftp.maltai.hu/Tanfolyam_2011_2012/Jegyzet/kapnografia.pdf
(2012.november)
- [26] Parkin Elmer: *TPS 2534* , Adatlap
- [27] MicroChip: *MCP3302/3304*, Adatlap
- [28] Apollo Union: *C20*, Adatlap
- [29] Freescale Semiconductor: *MPXV7007G*, Adatlap
- [30] Satronik Kft. – *Határértékek*
<http://www.satronik.hu> (2012. szeptember)

Függelék

A függelék a hardver kapcsolási és PCB rajzát tartalmazza, mely az Altium Designer által generált formában megtalálható a következő oldaltól.



1 2 3 4

A B C D

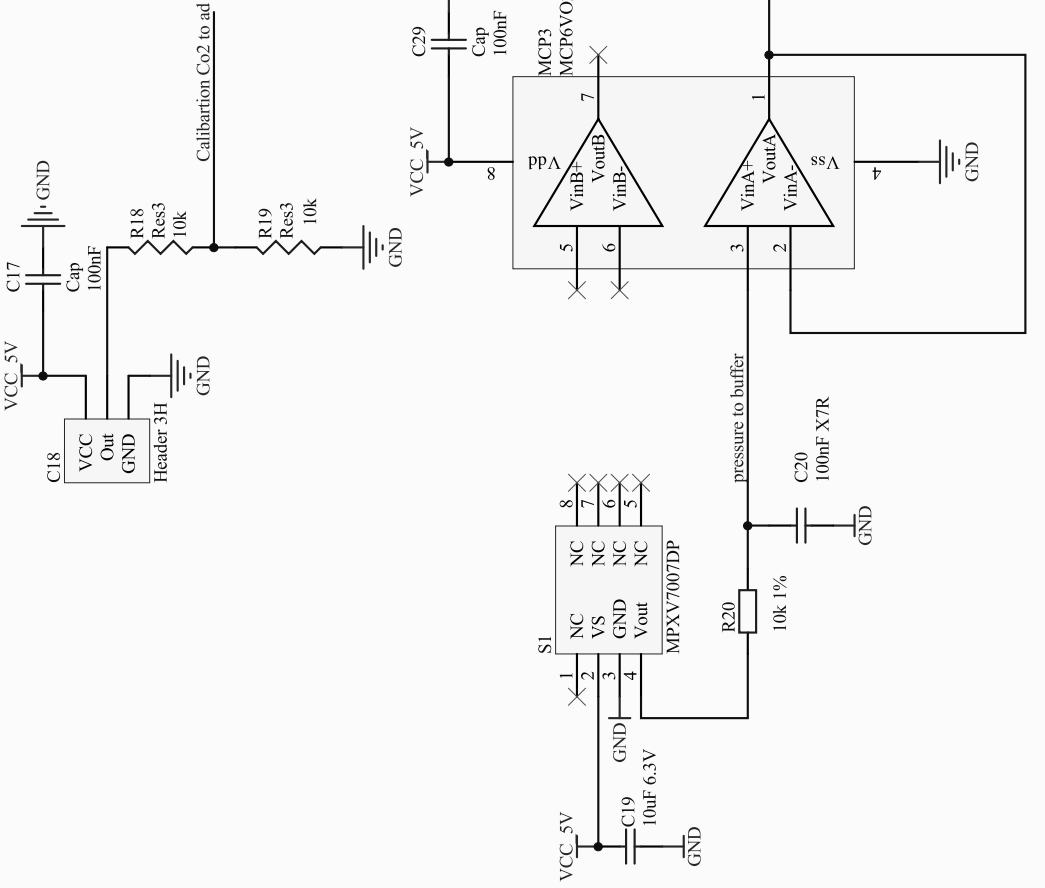
| | |
|-----------------------------|-----------|
| Title | |
| Size | Number |
| A | |
| Date: | Revision |
| 12/11/2012 | |
| File: | Sheet of |
| D:_PenDrive\...\CO2.SchDoc | Drawn By: |
| | |

A

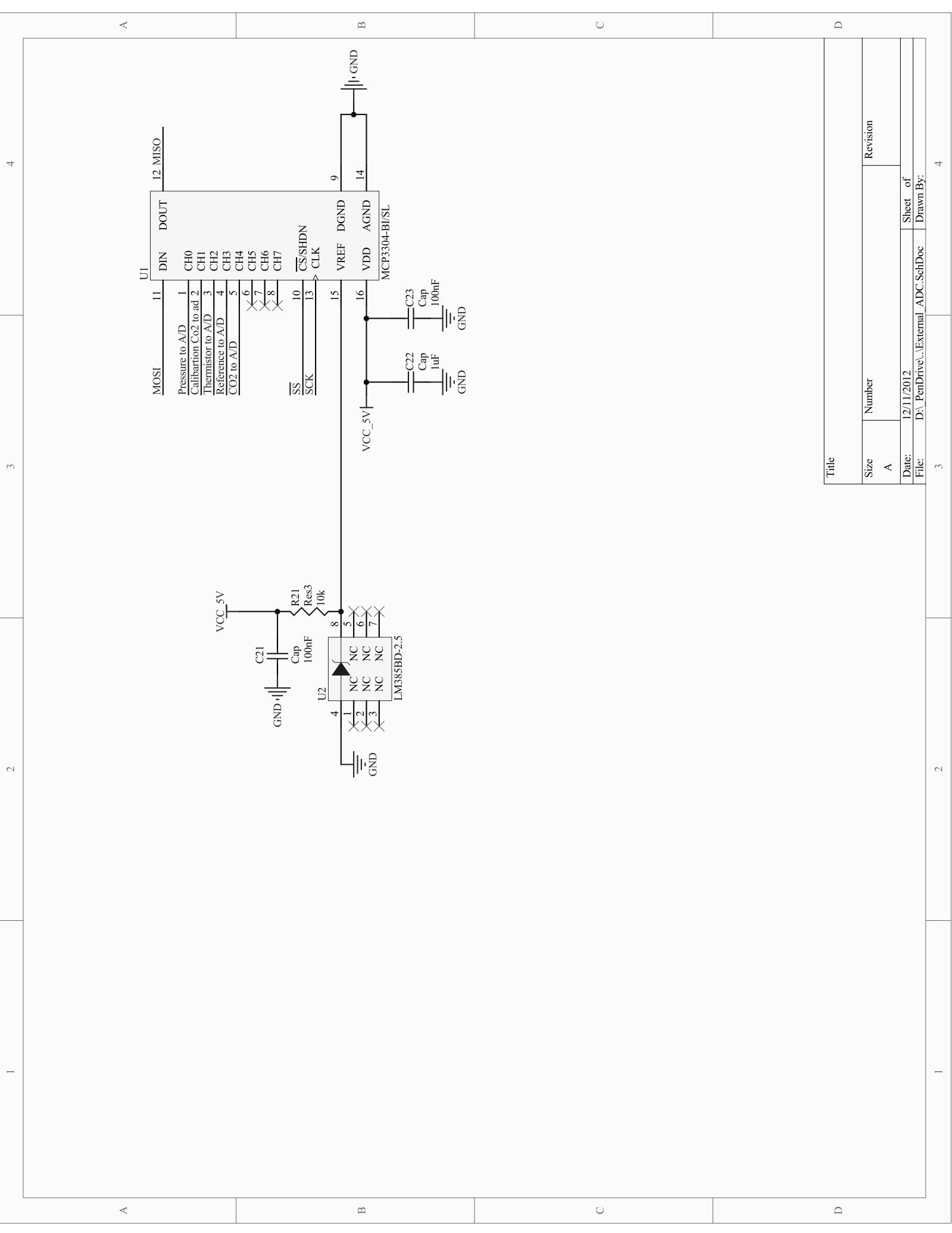
B

C

D

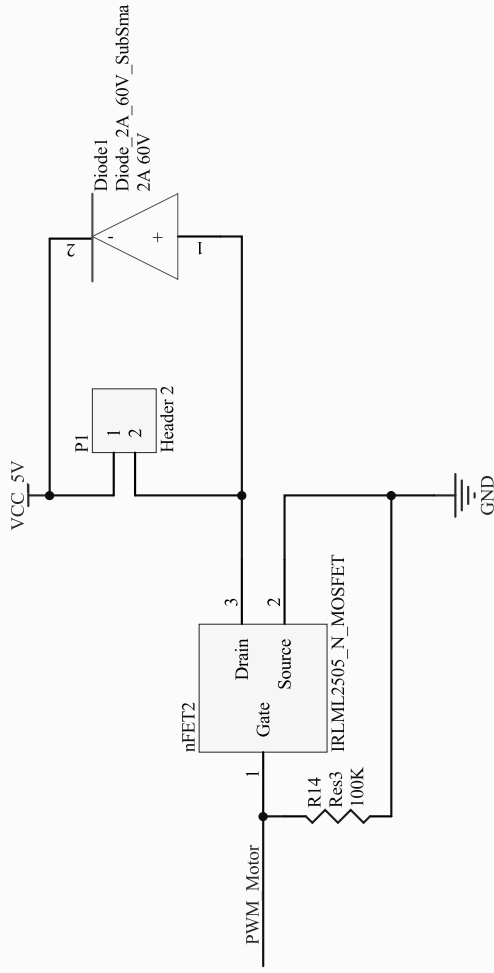
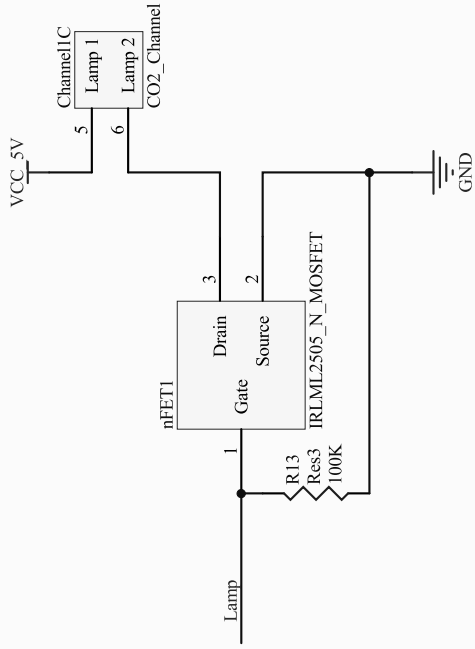


| | |
|-----------|---|
| Title | |
| Size | Number |
| A | |
| Date: | 12/11/2012 |
| File: | D:_PenDrive\...\ExtCo2_Pressure.SchDoc |
| Sheet of | 4 |
| Drawn By: | |



| MOSI | DIN | DOUT | MISO |
|-----------------------|-----|---------|------|
| Pressure to A/D | 1 | CH0 | 12 |
| Calibration Co2 to ad | 2 | CH1 | |
| Thermistor to A/D | 3 | CH2 | |
| Reference to A/D | 4 | CH3 | |
| CO2 to A/D | 5 | CH4 | |
| | 6 | CH5 | |
| | 7 | CH6 | |
| | 8 | CH7 | |
| | 10 | CS/SHDN | |
| | 13 | CLK | |
| | 15 | VREF | |
| | 16 | VDD | |

| | |
|-----------|--------------------------------------|
| Title | |
| Size | Number |
| A | |
| Date: | 12/11/2012 |
| File: | D:_PenDrive\...\External_ADC.SchDoc |
| Sheet of | 4 |
| Drawn By: | |



Title

Size

Number

Revision

Date:

12/11/2012

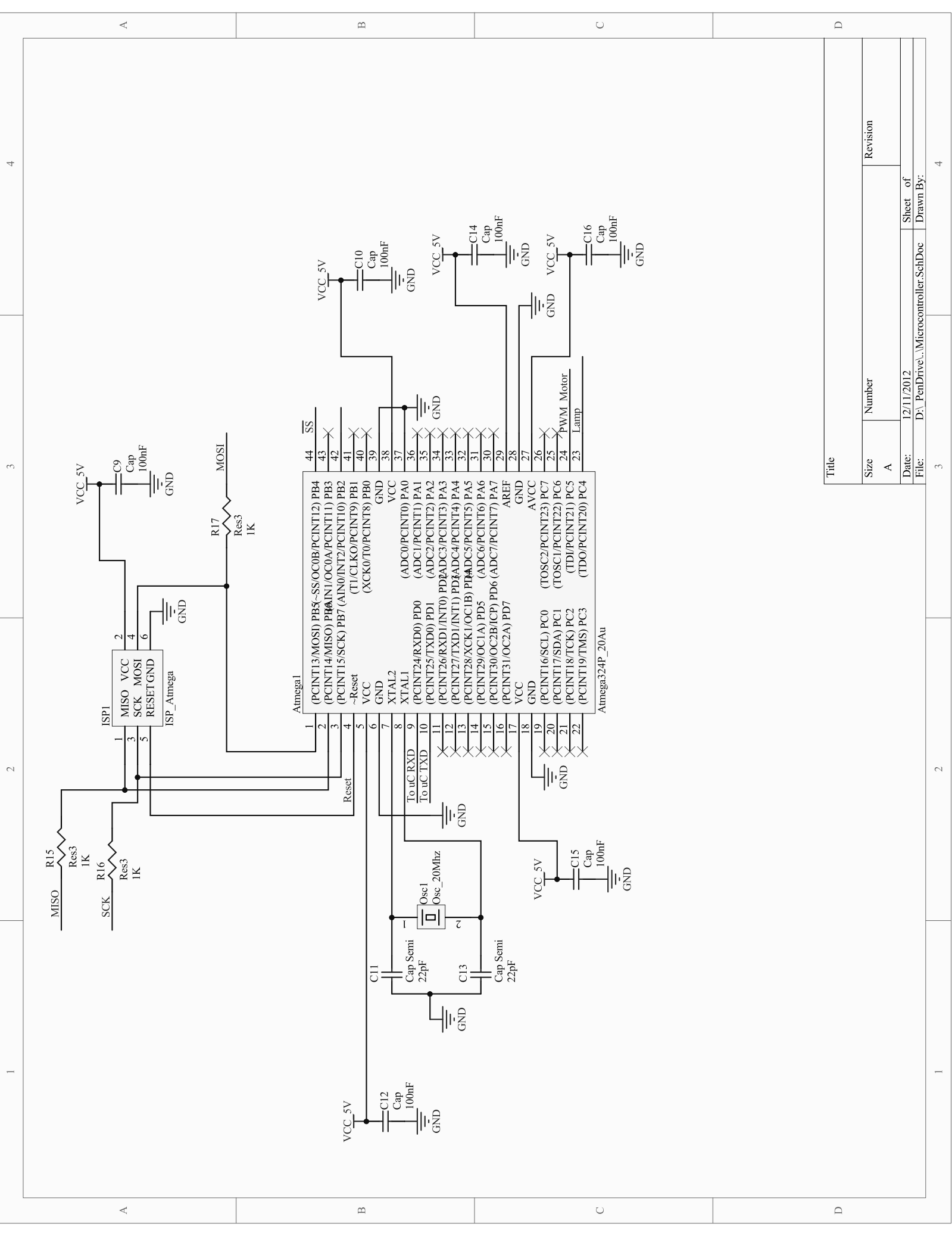
Sheet of

4

File:

D:_PentDrive\...\Lamp_Motor.SchDoc

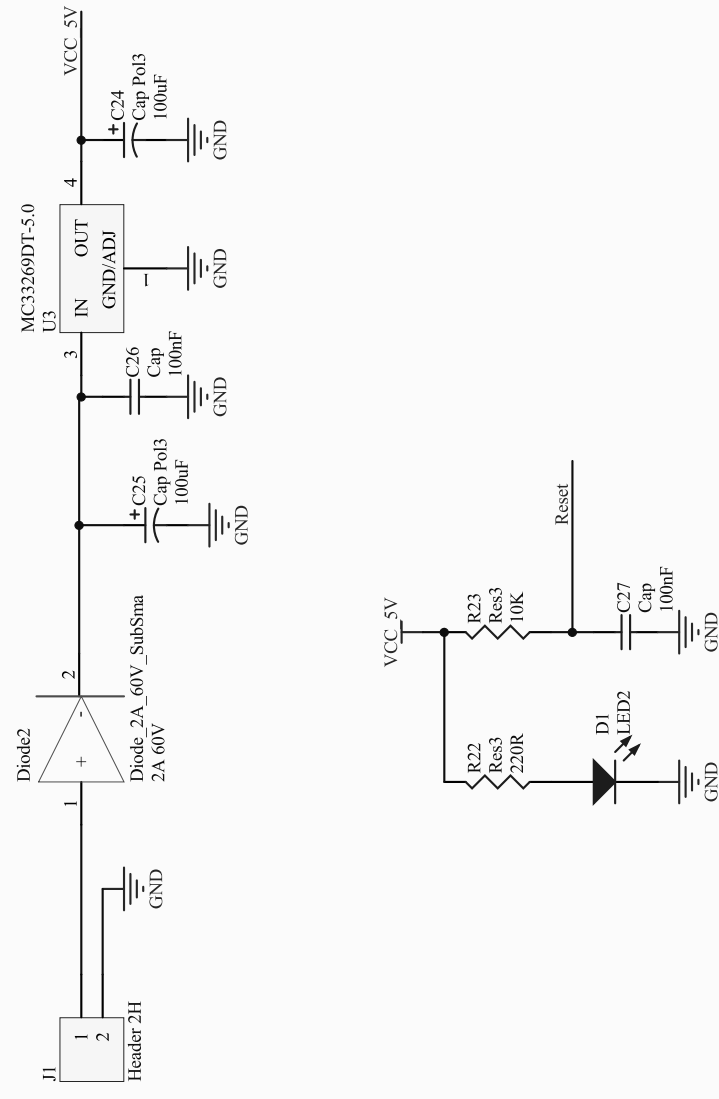
Drawn By:



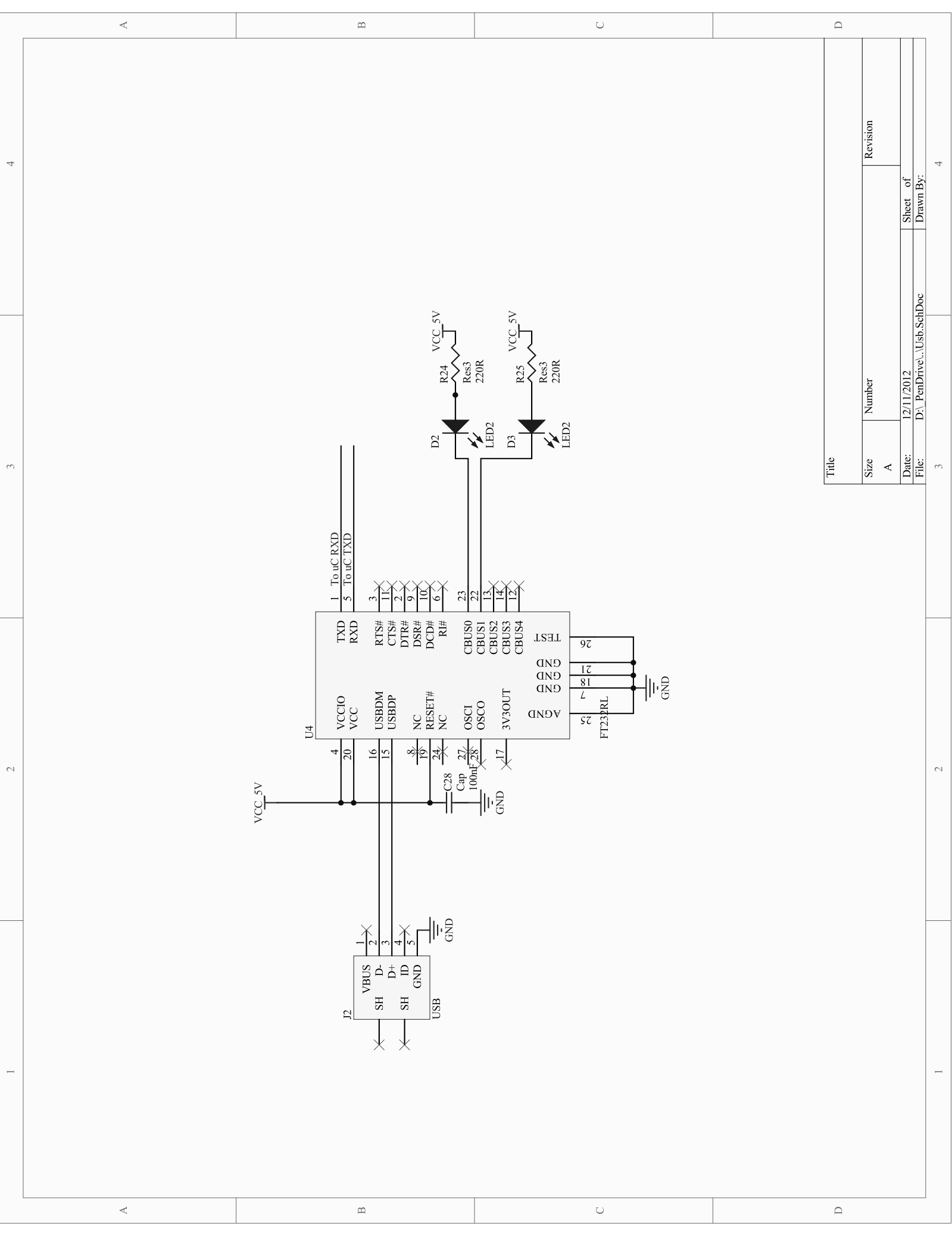
1 2 3 4

1 2 3 4

| | |
|-----------|--|
| Title | |
| Size | Number |
| A | |
| Date: | 12/11/2012 |
| File: | D:_PentDrive\...\Microcontroller_SchDoc |
| Sheet of | 4 |
| Drawn By: | |



| | |
|-------|-----------|
| Title | |
| Size | Number |
| A | Revision |
| Date: | Sheet of |
| File: | Drawn By: |



| | |
|-----------|-----------------------------|
| Title | |
| Size | Number |
| A | |
| Date: | 12/11/2012 |
| File: | D:_PenDrive\...\Usb_SchDoc |
| Sheet of | 3 |
| Drawn By: | |

Kalocsai David
CO2 Measurement
Ver. 1.0

