[Külső fekete borítólap formátuma]

Széchenyi István Egyetem
Gépészmérnöki, Informatikai és
Villamosmérnöki Kar

DIPLOMAMUNKA

Balogh Szabolcs József

Villamosmérnöki MSc szak

2015

[Gerincen:] Hallgató Neve, Évszám

SZÉCHENYI ISTVÁN EGYETEM
GÉPÉSZMÉRNÖKI, INFORMATIKAI ÉS VILLAMOSMÉRNÖKI KAR

DIPLOMAMUNKA

Müontomográfok kültéri hálózatának előkészítése vulkanikus tevékenység monitorozásához

Balogh Szabolcs József

Villamosmérnöki MSc szak
Automatizálási rendszerek szakirány

2015

Feladatkiírás

Hallgató adatai:
név: Neptun-kód:
szak: [szaknév] [MSc / MA / osztatlan / egyetemi]
specializáció:

Cím:
Feladatok leírása:

Belső konzulens adatai:

Név:

Munkahely:

Beosztás:

Külső konzulens adatai:

Név:

Munkahely:

Beosztás:

Győr, [feladatra jelentkezés dátuma]

_____ _____

[belső konzulens neve, beosztása] [tanszékvezető neve, beosztása]

Nyilatkozat

Alulírott, Balogh Szabolcs József (YQCDD4) Villamosmérnők MSc szakos hallgató kijelentem, hogy a „Müontomográfok kültéri hálózatának előkészítése vulkanikus tevékenység monitorozásához” című diplomamunka feladat kidolgozása a saját munkám, abban csak a megjelölt forrásokat, és a megjelölt mértékben használtam fel, az idézés szabályainak megfelelően, a hivatkozások pontos megjelölésével.

Eredményeim saját munkán, számításokon, kutatáson, valós méréseken alapulnak, és a legjobb tudásom szerint hitelesek.

Győr, 2015.12.10 _____

hallgató

Kivonat

Müontomográfok kültéri hálózatának előkészítése vulkanikus tevékenység monitorozásához

Diplomamunkám során bekapcsolódhattam az MTA Wigner Fizikai Kutatóintézet REGARD Csoport egy nemzetközi projektjébe. Ennek a projektnek a keretében magyar és japán kutatók közös célja egy olyan müntomográfokon alapuló rendszert építeni, amit vulkánok mellé telepítve képes azokat valós időben átvilágítani. A monitorozás során képet kaphatunk a magma mozgásáról, mely előre jelezhet vulkánkitöréseket. Ami a rendszerhez adott voltak magyar oldalról a müontomográfok, mint önálló mérő egységek, és japán oldalról a geofizikai szaktudás.

Diplomamunkámban megterveztem az online monitorozáshoz szükséges rendszert. A müontomográfokhoz GSM/GPRS modulokat illesztettem, programot írtam, melynek segítségével elküldhetik mérési eredményeiket egy távoli szerver felé. A több müontomográf aszinkron módon küldött eredményeit a szerveren MySQL adatbázisba menttettem, amihez a PHP programozási nyelvet választottam. Az online megjelenítéshez PHP, HTML, JavaScript programnyelvek együttes használatával fejlesztettem egy weblapot. Ez a honlap képes az adatbázisból megadott idő intervallumra megjeleníteni a müon fluxus adatokat, így vizualizálva a magma mozgását. Ezen felül mutatja a környezeti és detektor paraméterek időbeli változását, amivel vizsgálható a részegységek megfelelő működése. A fluxus adatok tekintetében képes összeintegrálni a programom több müontomográftól beérkezett mérési eredményt, és az így kapott adatokat megjeleníteni. Mindemellett feladatom volt egy kiegészítő hőmérséklet, nyomás, páratartalom mérőkártya kifejlesztése a tomográfhoz, melyet a tomográf mérőkörébe kellett illeszteni. A mért értékek befolyásolják a tomográf hatásfokát, illetve a magas páratartalom károsíthatja az elektronikát.

Abstract

Developing a system of muontomographs for volcano monitoring

I have joined the REGARD group of the MTA Wigner RCP where my thesis work became part of an international project. The collaboration of the hungarian and japanese researchers aim to develop novel cosmic muon based technology for volcanic research. During this imaging we get information on inner dynamics and possibility to forecast eruption. The joint project was based on the hungarian technic and the japanese expertise in geophysics**.**

In my thesis I focused to forging the individual detection elements into a system for online monitoring. I have integrated GSM/GPRS module to the muontomographs and wrote a program to send the results to the remote server. The asynchronously incoming data is saved into a MYSQL database on the server written in PHP. I have developed a web page for the online visualization, using the combination of PHP, HTML and JavaScript programming languages. Display of the muon flux data can be called for any time intervals as preparation for magma evolution studies. Furthermore the website can show the changes of the environment and detector parameters, usable for quality assurance and alert handling. My program is capable to merge flux data from the array of muontomographs and display the combined results. For the outdoor usage a temperature, pressure and humidity measuring board had to be developed and integrated into the tomographs data bus system, because ambient parameters influence the efficiency of tomograph and the in in extreme cases can damage the detector itself.

Tartalomjegyzék

1. Bevezetés ... 1
   1.1. Müontomográfia ... 2
   1.2. A Regard-Müontomográf felépítése és működése ... 3
      1.2.1. Detektor fizikai háttér ... 3
      1.2.2. Elektromos egységek ... 5
2. Rendszerterv ... 7
3. A THP mérőkártya ... 10
   3.1. Bevezetés ... 10
   3.2. Peremfeltételek ... 10
   3.3. Csatlakozás a müontomográf buszrendszeréhez ... 10
   3.4. Kiválasztott alkatrészek ... 11
      3.4.1. Microchip MCP9808 hőmérséklet szenzor ... 12
      3.4.2. Honeywell HIH613 021 001 páratartalom szenzor ... 12
      3.4.3. Bosch BMP180 nyomás szenzor ... 12
      3.4.4. Microchip PIC32MX170F256B mikrokontroller ... 13
      3.4.5. Microchip MCP1725 5V-3.3V lineáris feszültségszabályozó ... 13
   3.5. Kapcsolási rajz ... 13
   3.6. Nyákterv ... 16
   3.7. Programterv ... 17
4. A müontomográf GSM/GPRS kapcsolatának kiépítése ... 19
   4.1. Raspberry Pi ... 19
   4.2. GSM/GPRS modul ... 20
      4.2.1. Lábkiosztás és LED kijelzések ... 21
      4.2.2. Fogyasztás ... 23
   4.3. Kapcsolat a Raspberry Pi-vel ... 23
   4.4. GPRS kapcsolat kiépítése a Raspberry Pi –n ... 23
      4.4.1. További lehetőségek ... 24
   4.5. Mérési adatok küldése távoli szerverhez ... 24
5. Szerveroldali programozás ... 26
   5.1. Webszerver applikáció fejlesztése XAMPP ... 26
   5.2. Az alkalmazott programnyelvek és könyvtárak rövid bemutatása ... 26
      5.2.1. HTML ... 27
      5.2.2. MySQL ... 28
      5.2.3. PHP ... 29
      5.2.4. JavaScript ... 29
      5.2.5. Adat átadás a PHP és a JavaScript között: JQery, Ajax, JSon ... 29
      5.2.6. Megjelenítés – HighChars ... 30
   5.3. A webszerver mappa és fájlrendszerének a felépítése ... 32
   5.4. A Weblap felépítése ... 33
      5.4.1. A Weboldal megjelenítése böngészőben ... 33
      5.4.2. A weboldal HTML elemei ... 34
      5.4.3. A kliensoldali JavaScript függvények leírása ... 36
   5.5. MySQL adatbázis felépítése ... 37
      5.5.1. Teszt fájlok feldolgozása ... 37
      5.5.2. Adott RPi-hez tartozó hőmérséklet, páratartalom, nyomás, hatásfok és időbélyeg adatok tárolása ... 38

5.5.3. Adott RPi-hez tartozó fluxus adatok és időbélyeg tárolása 39
5.6. PHP szkriptek áttekintése 41
5.6.1. Adatbázis kezelő – DBHandler.php 42
5.6.2. Beérkező fájlok kezelése – IncomHandler.php 42
5.6.3. Lekérdezéskezelő – QueryHandler 45
5.6.4. Megjelenítés betöltő – PlotChart 45
5.7. Teszteredmények 46
Összefoglaló 47
Köszönetnyilvánítás 48
Irodalomjegyzék 49
Mellékletek 51
5.8. A kifejlesztett PHP, JavaScript, HTML programkódok 51
5.8.1. Index.php 51
5.8.2. QueryHandler.php 65
5.8.3. DBHandler.php 73
5.8.4. IncomHandler.php 74
5.8.5. PlotChart.php 80

1. Bevezetés

Diplomamunkám során alkalmam nyílt bekapcsolódni az Magyar Tudományos Akadémia (röviden MTA) Wigner Fizikai Kutatóintézet (röviden Wigner FK) REGARD (RMKI-ELTE Gaseous Detector Research and Development) Csoport egy alkalmazott fizikai projektjébe. Ez a projekt a vulkánok belső szerkezetének online vizsgálatáról szól, mely japán kutatók közreműködésével készül. A szükséges geológiai és geofizikai ismeretekekkel a japán kollégák rendelkeznek, magyar oldalról pedig a vulkán átvilágításához szükséges csúcstechnológiát képviselő müontomográf áll rendelkezésre. Nem ez az első eset, hogy müontomográffal vizsgálják vulkánok belső szerkezetét, de az eddigi tomográfok pontossága messze alulmarad a REGARD Csoport által kifejlesztettől. A projekt során egyszerre több müontomográfot fognak a terepen elhelyezni. Ennek a projektnek több mérnöki része még nem megoldott, ezek lesznek a diplomamunkám fő feladatai. A diplomamunkámnak nem része a müontomográf fejlesztése vagy a mérés és adatgyűjtés megvalósítása, ezek már rendelkezésemre állnak.

A dolgozatom témája egy online monitorozó rendszer megtervezése és megvalósítása, melyben a müontomográfok képesek a mérési eredményeiket eljuttatni a terepről egy távoli szerverhez. Így a célnak megfelelően létre kellett hoznom egy jól strukturált adatbázist, ahol a mérési adatok tárolása hosszú távon tartósnak, könnyen visszakereshetőnek és megjeleníthetőnek kell lennie. Az adatok külső helyről történő vizualizálásához a szerver és kliens oldali programok megírása és weblapfejlesztése is hozzátartozott.

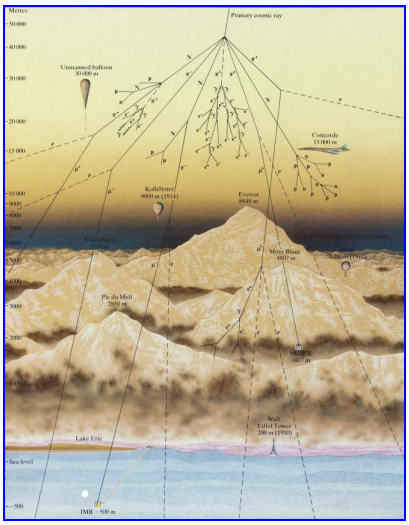
Emellett feladatom volt egy kiegészítő mérőkártya megtervezése, legyártása, felprogramozása is, mely képes hőmérsékletet, páratartalmat, nyomást (röviden THP-t) mérni, és illeszthető a tomográf busz rendszeréhez. Erre a kártyára a külső és belső környezeti változók regisztrálása miatt volt szükség. Ezen adatok birtokában tudjuk később korreláltatni például, hogy a tomográf hatásfoka hogyan függ a külső/belső hőmérséklettől, nyomástól, illetve a későbbiekben megfelelően korrigálni a detektor paramétereit és adatait.

Dolgozatom öt fő részre tagolódik. Az első bevezető fejezetben röviden írok a müontomográfiáról, annak fizikai hátteréről, emellett bemutatom a projekt során alkalmazott müontomográf felépítését és működését. A második fejezetben betekintést nyújtok a rendszer felépítésébe és tervébe, kihangsúlyozva azokat a feladatokat, amiket a diplomamunkám során nekem kellett megvalósítani. A harmadik fejezetben a kiegészítő, hőmérséklet, nyomás, páratartalom mérőkártya megtervezését és programjának működési elvét ismertetem részletesen;

kitérve a peremfeltételekre, alkatrész kiválasztásra, kapcsolási rajz, nyomtatott áramköri tervkészítésre, gyártásra és program felépítésre. A negyedik fejezetben a müontomográf mérőrendszerében lévő Raspberry Pi PC-t (röviden RPi) mutatom be. Részletesen kitérek arra, hogy hogyan lehet egy kiválasztott GSM/GPRS (Global System for Mobile Communications/ General Packet Radio Service) modult hozzáilleszteni a Raspberry Pi-hez, melynek segítségével mobilinternet hálózatra lehet kapcsolódni. Illetve a negyedik fejezetben ismertetem még az általam írt fájlfeltöltő programot, mely FTP (File Transfer Protocol) protokoll segítségével képes elküldeni a mérési adatokat a Raspberry Pi-ről egy távoli szerverhez. Az ötödik fejezetben a szerver oldali programok, könyvtárak rövid szemléltetése után áttekintést adok az adatbázis szerkezetéről is. Ebben a fejezetben bemutatom az általam létrehozott szerver és kliens oldali programokat, a megjelenítéshez használt weblap szerkezetét és ezek kapcsolatát.

1.1. Müontomográfia

Folyamatos kozmikus sugárzás éri földünket, a légkört kitevő oxigén és nitrogén atomokkal kölcsönható nagyenergiás részecskék hatására müonok keletkeznek a légkörben. Ezek a részecskék elérik a földfelszínt, áthaladnak rajtunk és a föld különböző rétegein is, energiájuktól függően különböző mélységig jutnak a földkéregben. [1] Az 1. ábrán látható, ahogy a légkörbe érkezve, egy kozmikus proton részecske záport indít.

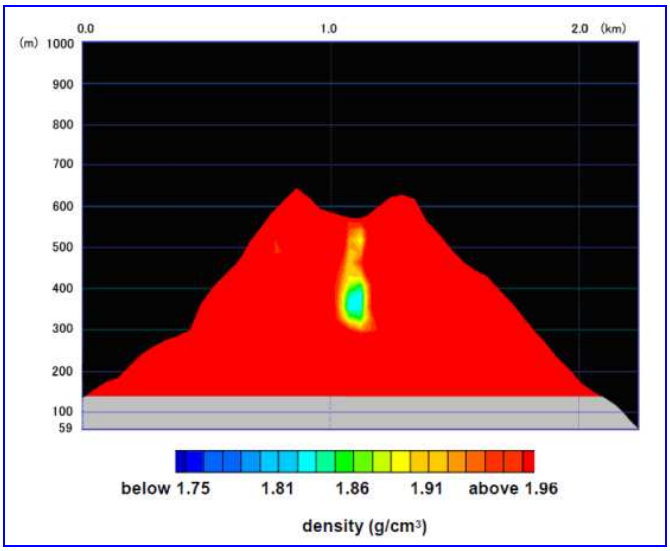


1. ábra – Sematikus példa egy kozmikus proton keltette részecskezáporra [1]

A kozmikus részecskéket a XX. század fordulóján fedezték fel. Az első olyan detektor megjelenéséig, amely gyakorlati alkalmazáshoz épült az 1960-as évekig kellett várni.

Ez volt az 1965-ben L.W. Alwarez által megépített müondetektor, mellyel a gízai Kefren-piramis belső szerkezetét vizsgálta, mérve a müonok beesési szögét, és az eloszlásukat. A müonok a különböző típusú anyagokban másképpen nyelődnek el, ezáltal rekonstruálni tudta a környező anyagmennyiséget, sűrűséghosszat. Méréseivel megállapította, hogy nincs ~2m-nél szélesebb üreg a piramisban így azt nem kellett a feltételezett rejtett kamra kereséséhez megbontani [1] [2]. Ez volt az első alkalmazása a kozmikus müonoknak.

A kozmikus müonok felhasználásának egy másik fontos területe a vulkánok belső szerkezetének feltérképezése. Ez az 1990-es években kezdődött, amikor K. Nagamine és kutatócsoportja a Tsukuba vulkán sűrűségeloszlását határozta meg müontomográf segítségével [27]. A folytatásban H. Tanaka térképezett fel több vulkánt is müontomográfok segítségével, a kezdeti 30%-os pontosságot mára már nagymértékben tudták javítani [28]. A 2. ábrán H. Tanaka kutatócsoportja által vizsgált Satsuma-Iojima vulkán sűrűség eloszlás képét láthatjuk, melyet müontomográf felhasználásával készítettek.



2. ábra - a Satsuma-Iojima vulkán müontomográffal meghatározott sűrűségeloszlása [28]

1.2. A Regard-Müontomográf felépítése és működése

1.2.1. Detektor fizikai háttér

Gáztöltésű detektoroknál a detektálni kívánt töltéssel rendelkező részecske a gázon való

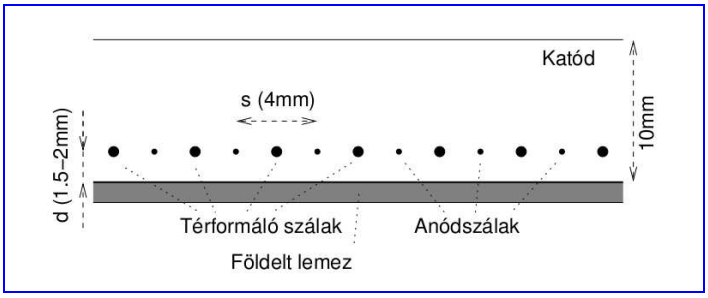
áthaladáskor elektromos terével gerjeszti a gáz atomjait, ami mellett elhalad. Ekkor két dolog történhet, vagy több energiát közölt, mint a gázatom elektronjának kötési energiája, ekkor kiszakít elektront, vagy ha kevesebbet, akkor csak gerjeszti őket, melyek a többlet energiát pl. nemes gáz esetén UV foton kibocsájtással vezetik le. A gázban történő áthaladáskor tehát keletkeznek elektronok és UV fotonok. Az elektronok száma azonban még nem elegendő ahhoz, hogy detektálni tudjuk őket. Ahhoz, hogy mérhető villamos jelet kapjunk, több elektronra van szükségünk. Ezt pl. úgy érhetjük el, hogy a gázt nagy elektromos térerősségbe helyezzük, ahol az a kevés létrejött elektron felgyorsul, melyek aztán újabb elektronokat szakítanak ki a gáz atomjaiból, mígnem egy elektron lavina jön létre [29]. Ilyen elven működik a Geiger-Müllerszámláló is.

A Close Cathode Chamber (CCC, közeli katódos kamra) alapú müontomográfot az MTA WignerFK REGARD csoportjában Dr. Varga Dezső, Dr. Hamar Gergő és Dr. Oláh László fejlesztették ki. Az általuk kidolgozott kamrákkal egy költséghatékony, mozgatható, kiváló hely és szög felbontású müontomográfot építettek [30][31].

A CCC konstrukcióban, a kamrákban Ar-CO_2 gázkeveréket alkalmaznak 80:20 arányban [2]. Az argon mellett több érv is szól. Fontos tulajdonságai, hogy kémiailag semleges, olcsó, kevés szabadsági foka van és viszonylag nagy a rendszáma. A kémiai semlegesség fontos, mert így nem teszi tönkre a detektor belsejét. Az alacsony ár, magáért beszél. A kevés szabadsági fok azért hasznos, mert így szinte minden gerjesztési energia az elektronoknak adódik át, nem pedig pl. fogásra vagy pulzálásra fordítódik. A nagy rendszámú elemek több elektronnal rendelkeznek, külső elektronhéjukon pedig kisebb a kötési energia, így kisebb energia hatására is kilépnek a kötésből az elektronjaik. Felmerül a kérdés, hogy ha ennyire kedvező tulajdonságú az argon, akkor miért van szükség CO_2 gázra? A válasz abban rejlik, hogy a gáz gerjesztése során UV fotonok is keletkeznek. Ezek az UV fotonok a detektor réz vagy arany fémfelületét elérve annyi energiát tudnának közölni, hogy onnan ki tudnának szakítani további elektronokat, amiket aztán a villamos tér felgyorsítana, és további elektronokat szakítanának ki a gázatomoktól, ami újabb elektron lavinához és UV sugárzáshoz vezetne. Az így pozitívan visszacsatolt rendszerben folytonos elektron lavina lenne, ami tönkretenné a detektort. Ennek az elkerülésére kellett egy plusz gázt is alkalmazni a CO_2 –t, ami el tudja nyelni az argon által kibocsájtott UV fotonokat. Az ilyen gázokat kioltó gázoknak is nevezik [29].

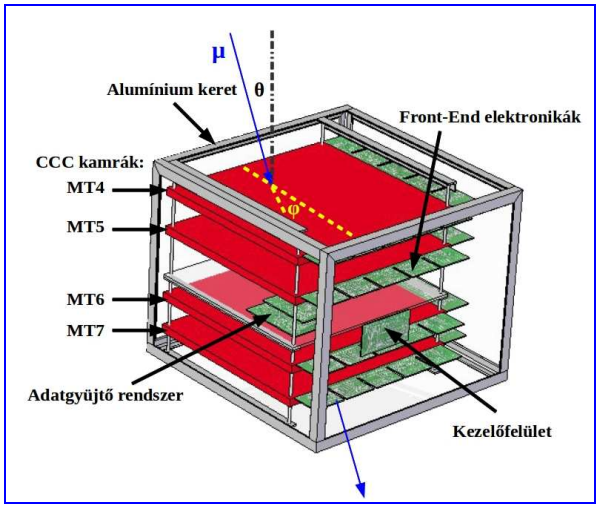
A CCC kamrákban két féle huzalt, egy térformáló 100µm- és érzékeny 21µm átmérőjű szálat alkalmaznak. A szálak alatt 1,5mm-re a szálakra merőlegesen 4mm x 320mm katód réz pad-ek, réz parketták helyezkednek el. Villamos jel mérhető az anódon szálon, a térformáló szállakon,

valamint a katódon [2]. A CCC elrendezés előnye, hogy a szálakat nem szükséges erősen rögzíteni, így sokkal könnyebb keretek elegendőek a szállak megtartásához. További előnyök, hogy a rendszer érzéketlen a kis mechanikai deformációkkal szemben, illetve könnyű szerkezetű és költséghatékony [30][31]. A 3. ábrán egy CCC kamra felépítésének sematikus ábrája látható.



3. ábra - A CCC kamara belső szerkezete, benne az elektromos teret meghatározó anód, katód és térformáló szálakkal és föld lemezzel [2]

Több ilyen kamrából álló réteg (minimum 2) szükséges ahhoz, hogy egy müon részecske pályája meghatározható legyen, több réteg alkalmazása növeli a pontosságot és a detektálási valószínűséget. A 4. árán egy a REGARD csoport által fejlesztett négyrétegű müontomográf sematikus ábrája látható.



4. *ábra – Egy négyrétegű müontomográf szerkezeti felépítése [2]*

1.2.2. Elektromos egységek

A tomográf által detektál jeleket automatizált adatgyűjtő rendszer olvassa ki. A tomográf és az adatgyűjtő rendszer fő elektromos egységei és azok funkciói a következők:

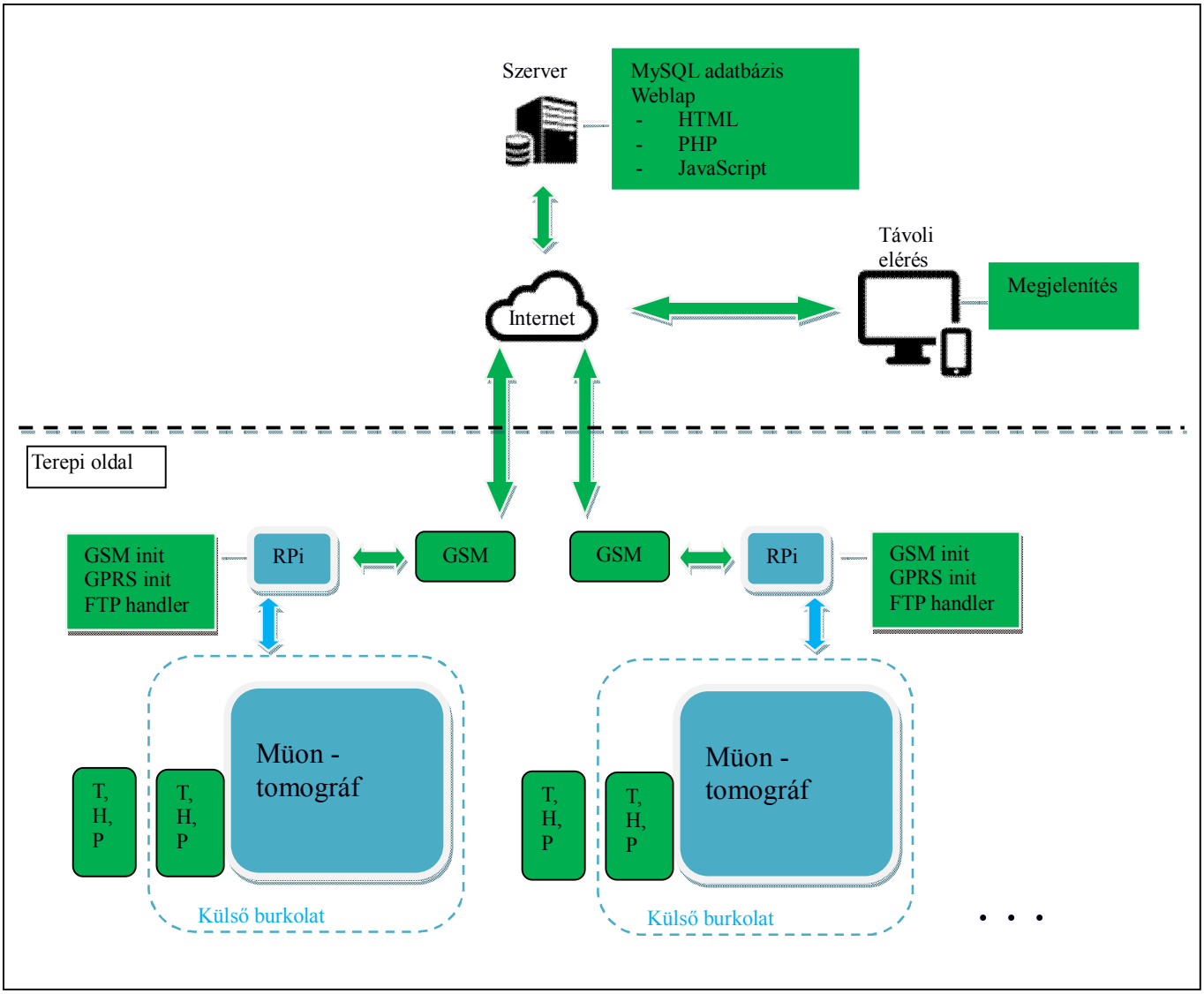
- akkumulátor (12V): a rendszer áramellátását biztosítja a terepen,
- nagyfeszültségű tápegység: a tomográf kamráinak nagyfeszültségű tápellátását biztosítja,
- trigger elektronika: egy külön fejlesztett gyors hardver, ami a részecskék berkezését figyeli,
- front end elektronikák: a térformáló szálakról és a pad-ekről olvassák be az elektromos jeleteket shiftregisztereikbe, az áthaladási pozíció meghatározásához,
- vezérlő egység: ez egy Raspberry Pi PC, ez felel az adatok kiolvasásáért, lementéséért és kiértékelésért

A CCC-ben az elektromos jeleket három forrásból olvashatjuk ki, az anód szálról, a térformáló szálakról és a pad-ekről. Az anód szálak az egész detektorban össze vannak kötve, ezt használják arra, hogy triggerelje/indítsa a mérést. Egy külön elektronikát fejlesztettek ehhez, mely legalább három trigger esemény hatására jelzi a vezérlőnek, hogy adja ki az olvasási parancsot a front end elektronikáknak. A front end elektronikák kiolvassák az egyes szálak és parketták jeleit és letárolják azokat sorba kötött shift regisztereikbe. Ezután a shift regiszterek érékeit beolvassa a Raspberry Pi PC, és lementi azok értékeit, a beolvasott nyers adatokra pedig lefuttatható a Raspberry Pi-n lévő kiértékelő algoritmus. A tomográf kiegészítő szenzorhálózata is, shiftregiszterekben tárolja az információt. A THP mérőkártyát úgy kell megtervezni, hogy ebbe az adatfolyamba be tudjon kapcsolódni.

2. Rendszerterv

A magyar-japán együttműködés fő feladata vulkánok online monitorozása müontomográfok segítségével, amivel előre tudunk esetleges vulkánkitöréseket jelezni. Ahhoz, hogy a monitorozásban viszonylag rövid idő alatt képet kaphassunk a magma állapotáról, nagy felületű detektort kell alkalmazni. Ezt részben fizikai korlátok miatt, nem egy nagy detektort szállítunk a terepre, hanem több kisebb felületűt. A jó hely és szög felbontású müontomográfok tehát adottak a terepi oldalon, melyek autonóm mérési és kiértékelési rendszerrel rendelkeznek. Ahhoz, hogy a tomográfok eredményeit online tudjuk monitorozni első feladatom volt, ezeket hálózatra kötni, hogy egy közös szerveren tárolhassuk az adataikat. A közös adattárolás azért is kell, mert bár külön-külön mérnek, de a vulkánhoz képest pontszerűnek tekinthetjük őket, ezért részecske fluxus adataik összeadhatók. Így hamarabb megkapjuk vulkán sűrűség képét, ezáltal a magma állapotot.

Az 5. ábrán látható az általam megtervezetett rendszer fő hardveres és szoftveres elemei.



5. ábra – A vulkán online monitorozó rendszerének fő elemi

Az 5. ábrán zöld színnel szerepelnek azok az elemei a rendszernek, melyeket a diplomamunkám során megvalósítottam. A kékkel színezett részek már elkészültek vagy nem én fejlesztettem őket. A három pont azt szimbolizálja, hogy nem csak kettő, hanem több müontomográf is lehet a rendszerben, melyek autonóm működnek. A szerveroldalt felkészítettem ezek aszinkron működésére.

Az ábrára nézve három fő munkaterületet lehet jól elkülöníteni. Az első a müontomográfhoz tartozó hőmérsékletet (T), páratartalmat (H), nyomást (P) mérő kártya, melyet megterveztem, legyártattam, megterveztem a programjának szerkezetét, mindent úgy, hogy integrálható legyen a tomográf adatbusz rendszerébe. A második terület a Raspberry Pi-n lévő feldolgozott mérési

eredmények eljuttatása egy távoli szerverhez. Ebben a folyamatban egy GSM modult használtam, mely képes GPRS kapcsolatra. Ahhoz, hogy kiépüljön a kapcsolat a Raspberry Pi-n szükséges volt a GSM modul megfelelő hardveres és szoftveres illesztése. Ennek megvalósítása érdekében egy külön programot írtam, ami FTP segítségével, internetkapcsolaton keresztül elküldi egy szervernek a mérési eredményeket. A harmadik fázis a szerveroldali programozás volt. A szerveren a mérési eredményeket MySQL adatbázisba töltöttem fel, PHP (Hypertext Preprocessor) segítségével. A webes elérésben vegyesen használtam PHP, JavaScript és HTML (HyperText Markup Language) nyelveket. A kliens oldalnak már csak a megjelenítéshez szükséges adatokat adtam át. A grafikonok kirajzolásához egy önálló JavaScript könyvtárat használtam.

3. A THP mérőkártya

3.1. Bevezetés

A THP mérőkártya igénye a diplomamunka téma megbeszélések alatt merült fel. A kísérletek során szeretnék pontosan látni és lementeni a környezeti változókat (hőmérséklet (T), nyomás (P), páratartalom (H)) mind a berendezés belsejében, mind a külsejében. Ezek az adatok nagyon fontosak, a mérések során a hőmérséklet és nyomás értékektől függ a detektor hatásfoka [29]. A páratartalom pedig hatással lehet, az elektronikai egységekre.

3.2. Peremfeltételek

A WignerFK REGARD Csoporttól, mint megrendelői oldaltól az alábbi igények érkeztek be a THP kártyára vonatkozóan:

- T szenzor pontosság: 0.5 °C,
- P szenzor pontosság: 1mBar,
- H szenzor pontosság: 5% RH,
- a mérőkártyának csatlakoznia kell a tomográf belső adatbusz hálózatára,
- alkalmasnak kell lennie sorozatgyártása,
- költséghatékony kell, hogy legyen
- minél kisebb legyen a fogyasztása

3.3. Csatlakozás a müontomográf buszrendszeréhez

Egy 9-es tüskesorral lehet a tomográf buszára csatlakozni. A tomográf busz rendszere sorosan kapcsolt 74HCT165 típusú shiftregiszterekkel van kialakítva. A kiolvasó órajelet egy Raspberry Pi GPIO lába adja, ami jelenleg 2MHz. Erre a bitfolyamra kell rácsatlakozni úgy, hogy az nem feltétlenül végpont lesz, tehát előtte és utána is lehet shiftregiszter.

Azt a megoldást választottuk, hogy a szenzorokat egy mikrokontroller buszrendszerére fogjuk kötni, mely periodikusasan olvassa a mért értékeket. Ha adat kiolvasás történik, akkor a mérőkártya szimulálja, hogy ő egy 3 sorosan kapcsolt shiftregiszter, ami a következőképpen valósul meg:

A THP kártya mellett lévő shiftregiszternek, ami várja az adatokat, elküldi sorban a három szenzor

mért értékeit bitesen. A kártya másik oldalán lévő shiftregisztertől, ami küldi a kontrollernek a biteket, folyamatosan beolvassa és lementi az adatokat. Ha már elküldte a három szenzor mért értékékeit, akkor következik a lementett bitek továbbítása.

3.4. Kiválasztott alkatrészek

Az alkatrészek kiválasztása során a fenti szempontokon túl a következőket tartottam szem előtt:

- az ár lehetőleg alacsony legyen,
- leinformálható, neves cégek termékeinek jártam utána,
- megbízható beszállítókat kerestem,
- elsősorban nagy hőmérséklettartományon működő szenzorok érdekeltek,
- kis tokozású IC-ket és kisméretű alkatrészeket kerestem,
- alacsony fogyasztású szenzorok után kutattam,
- fontos volt, hogy a szenzorok mérési karakterisztikái a hőmérséklet függvényében lineárisak és hiszterézis-mentesek legyenek, lehetőleg az egész hőmérséklettartományban vagy annak nagy részén,
- a szenzorok adatbuszra kapcsolhatóak legyenek, elsősorban I2C vagy SPI buszra.

A mikrokontroller kiválasztásánál pedig az alábbi szempontok voltak meghatározóak:

- az alkatrész általános célú,
- kevés perifériájú,
- kis lábszámú,
- 32 bites eszköz legyen,
- közepes programmal és adat memóriával rendelkezzen,
- viszonylag nagy sebességgel bírjon a tomográf adatbusz órajeléhez képest (40-50MHz),
- beépített órajel generátort tartalmazzon,
- valamint megbízható, neves gyártó terméke legyen.

Diplomamunkám következő alfejezeteiben a kiválasztott alkatrészekről adok egy rövid,

átfogó képet.

3.4.1. Microchip MCP9808 hőmérséklet szenzor

A kiválasztott hőmérsékletszenzor a -20 °C – 100 °C tartományon belül +/- 0.5 °C maximális hibával képes hőmérsékletet mérni. A tápfeszültsége 2.7 V – 5.5 V között választható meg. Fogyasztása elenyésző, tipikusan 200 μA, de még ez is csökkenthető az üzemmód változtatásával. Tokozása MSOP-8 nagyon kisméretű. Az IC-n belül ΔΣ egység végzi az analóg – digitális átalakítást. Busz kommunikációs lehetőségei SPI vagy I^2C. A szenzorban található kontroll regiszterek segítségével lehetőség nyílt beállítani a mérési pontosságot, programozhatunk Shutdown vagy Low-power üzemmódokat, beállíthatunk Alert hőmérséklet szinteket, melyek aktiválják az IC Alert lábát. Ezekből a funkciókból csak a mérési pontosság beállításának lehetőségét használtam, de későbbi továbbfejlesztésekhez pl. a fogyasztás optimalizálás vagy vészleoldás céljából még jól jöhetnek.[3]

3.4.2. Honeywell HIH613 021 001 páratartalom szenzor

A kiválasztott páratartalom szenzor -25 °C és 85 °C között képes relatív páratartalmat mérni. Az eszköz hőmérséklet kompenzáltan mér 10 % és 90 % között +/- 4 % hibával. Beépített hőmérséklet szenzorral rendelkezik, ami viszont esetünkben nem használható, mert +/- 1 % hibával mér, ami nem felel meg a 3.2. fejezetben leírt feltételeknek. Tápfeszültsége 2.3 V – 5.5 V között változtatható, fogyasztása alacsony 0,6 mA – 0,8 mA között változik. Lehetőség van sleep üzemmódot alkalmazni, ha tovább szeretnénk csökkenteni a fogyasztást, de munkám során nem használtam ezt a funkciót a folyamatos kiolvasás miatt. I^2C vagy SPI buszon tud kommunikálni. Ez a szenzor is programozható Low/High Alert jelzésére, mely ugyan nem kapcsolódik közvetlenül a diplomaunkámhoz, ám a későbbi fejlesztéseknél kiválóan alkalmazható lesz a nagyfeszültség biztonsági lekapcsolásához. Tokozása SOIC-8. [4]

3.4.3. Bosch BMP180 nyomás szenzor

Az alkatrész nagyon kisméretű LGA tokozású. Tápfeszültsége 1.8 V – 3.6 V, 5

μA/mintavétel fogyasztású. Nagy felbontású, 300 hPa-tól 1100 hPa-ig képes mérni. Relatív hibája 0.12 mBar, abszolút hibája tipikusan 1 mBar. Beépített hőmérséklet szenzorral rendelkezik, de ennek pontossága nem megfelelő a 3.2 fejezetben megfogalmazott pontossági követelményhez. Kommunikációs interfészei I^2C és SPI. [5]

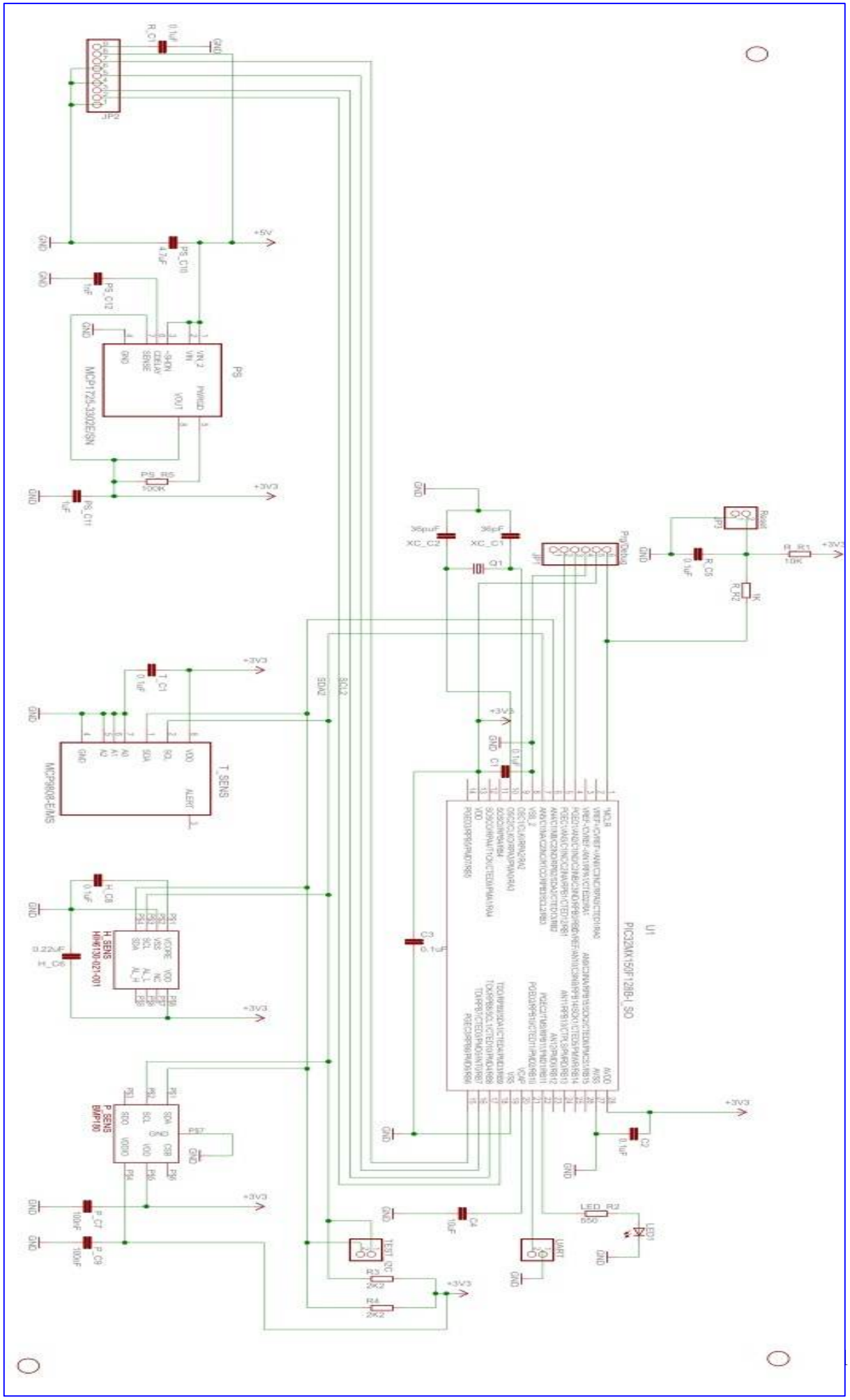
3.4.4. Microchip PIC32MX170F256B mikrokontroller

A mikrokontroller főbb jellemzői: maximum 50 MHz órajel, MIPS32 M4K CPU-val rendelkezik, -40 °C és 85 °C kötött üzemeltethető, 28 lába van, 256 KB flash ROM-mal és 64 KB SRAM-mal rendelkezik, számos perifériával rendelkezik, többek között I^2C busz interfésszel is, amire a szenzorokat csatlakoztattam a mérőkártyán. A Microchip honlapján található egy kontroller kiválasztást segítő szolgáltatás. Itt a kívánt memória méretre, tokozásra, perifériákra stb. be lehet állítani szűrőket, e peremfeltételek kitöltve ajánlotta ezt a kontroller. [6]

3.4.5. Microchip MCP1725 5V-3.3V lineáris feszültségszabályozó

Ez egy egyszerű feszültségszabályozó, alacsony fogyasztással. Maximálisan 500 mA-t tud leadni a kimenetén. Erre az alkatrészre azért volt szükség, mert a buszról 5V-os tápellátást kaptam, de a kártyán lévő alkatrészek 3.3V-os feszültségszinten működnek. Van belőle változtatható vagy fix kimenetű, az utóbbit választottam, mert kevesebb alkatrész kell hozzá és mert nem lesz szükség később a feszültség szint változtatására. [7]

3.5. Kapcsolási rajz



6. ábra – A THP kártya kapcsolási raja

A kapcsolási rajzot, ami a 6. ábrán látható, és a nyáktervet is Eagle Cadsoft (v6.4.0) programmal rajzoltam. A program adatbázisában nem szereplő új alkatrészekhez tartozó Eagle-filet az adatlapok alapján magam szerkesztettem meg. Ilyenek voltak például a BMP180-as és a HIH613-as IC-k. A Microchip termékek esetében annyival volt könnyebb a dolgom, hogy a gyártó weboldaláról letölthető egy alkatrész kapcsolási rajz és nyákterv generátor tool. Ezt a toolt az Accelerated Designs céggel közösen hozták létre, a céljuk az volt, hogy a Microchip alkatrészek kapcsolási rajzát és NYÁK tervét minden nevesebb tervezőprogramhoz le lehessen generálni. Jómagam is ennek a toolnak a segítségével generáltam le Microchip alkatrészek kapcsolási rajzait és NYÁK terveit tartalmazó lib-fájlokat, melyeket aztán az Eagle programban könnyedén fel tudtam használni munkám során.

A kapcsolási rajznál törekedtem az elnevezésekben jelezni, hogy mely ellenállás, kondenzátor stb. mely IC-hez tartozik.

A kártya szenzorhálózatához kiválasztott busz kommunikáció az I^2C lett, mert minden szenzor rendelkezik ilyen interfésszel, kevesebb vezeték szükséges hozzá, és nincs szükség nagyon gyors kommunikációra.

A prototípust a hibakeresés segítése érdekében extra elemekkel láttam el, ilyenek az UART kommunikáció, I^2C busz kivezetés, teszt LED, és egy kvarckristály.

A LED1-en kivilloghatunk a programból debug információkat, ez elég kezdetleges megoldás, de hasznos.

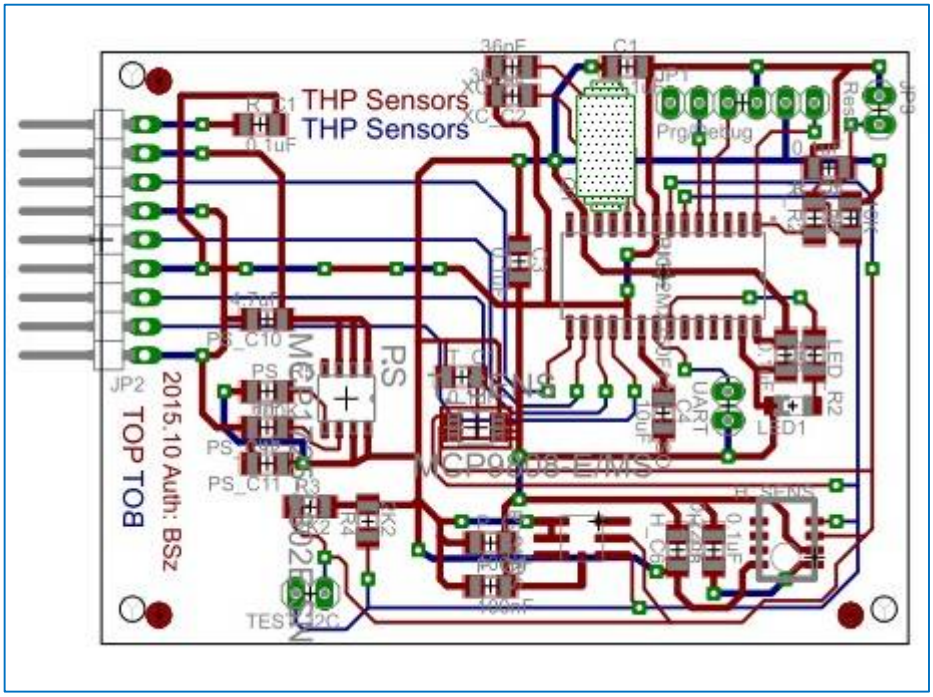
Mikor fut a programunk a kontrolleren, sajnos nincs konzol, amire pl. printf() függvénnyel kiírhatnánk a debug információkat. Erre nyújt megoldást az UART tüskesor, csak a GND és a kontroller felöli Tx láb van rákötve. A kontrollerben csak inicializálni és konfigurálni kell az UART-ot, majd a debug információkat futás közben ide kiírhatjuk. A tüskesort megfelelő illesztéssel hozzácsatlakoztatva PC-hez, terminálból olvashatjuk, hogy milyen üzeneteket kapunk a kontrollertől.

A TEST_I2C tüskesor segítségével belehallgathatunk az I^2C kommunikációba, ami szintén a hibakeresést könnyítette meg a munkám során. A Q1 kristály használata nem kötelező, de ha később úgy döntenénk, hogy pontosabb órajelre van szükségünk a megfelelő kommunikációhoz, akkor könnyedén beállíthatjuk a kontrollerben külső órajel forrásként.

A kapcsolási rajzon található egy 9-es tüskesor, ezen keresztül kell csatlakozni a tomográfra. A kapcsolási rajz szerint a 9-edik lábtól az első láb felé haladva a funkciói rendre a következők: Threshold, 5 V, Trigger, GND, Clock, GND, DataIn, DataOut, GND. A Thresholdra nem volt

szükségem, ezt egy kondenzátorral a földre kötöttem. Az 5 V és GND a feszültségszabályzóra került. A többi láb 5 V-os jelszinten dolgozik, ezért azokat a kontroller 5 V toleráns lábaira kötöttem.

3.6. Nyákterv

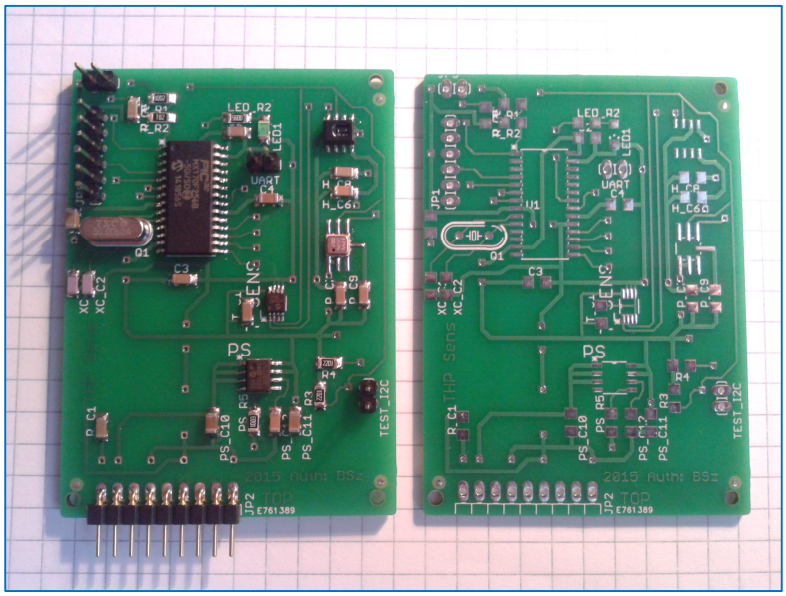


7. ábra – A THP kártya nyákterve (1,5x nagyítás)

A THP mérőkártya nyákterve a 7. ábrán látható. A nyákot kétoldalasra terveztem, próbáltam minél kisebb helyen elhelyezni az alkatrészeket. Az egyes alkatrészeket (ellenállás, kondenzátor… stb.) az adott IC-khez csoportosítottam, melyekhez tartoznak. A hidegítő kondenzátorokat igyekeztem minél közelebb elhelyezni a V_{DD} és GND pad-ekhez. A fő vonalak (táp és földvezetékek) 24 mil (0,05 inch = 1.27 mm = 50 mil) szélességűek, a jelvezetékek 12 mil szélességűek. Ezt mindenhol próbáltam tartani, ahol lehetett, de néhány pad olyan keskeny volt, hogy muszáj volt fő vezetékek esetében kisebbet használni. A mechanikai rögzítéshez elegendő lehet a 9-es tüskesor, de elhelyeztem 3 furatot a 3 sarokban, ha csavarral való rögzítésre kerülne sor. A beültető gépek jobb pozícionálásához elhelyeztem három fiduciális jelet, melyekhez viszonyítva helyezhetik le az alakrészeket a nyákra.

Prototípus lévén nem alkalmaztam „föld folyatást” avagy „föld szigeteket” annak érdekében, hogy ha esetleg át kell kötni valamit a kártyán, akkor viszonylag könnyen el lehessen végezni. A prototípuson a 9-es tüskesornál a 2.54-es tüskesort használtam az 1.27-es helyett az egyszerűbb hibakeresés érdekében, ezért a prototípus esetén szükség volt egy kis átalakító elkészítésére.

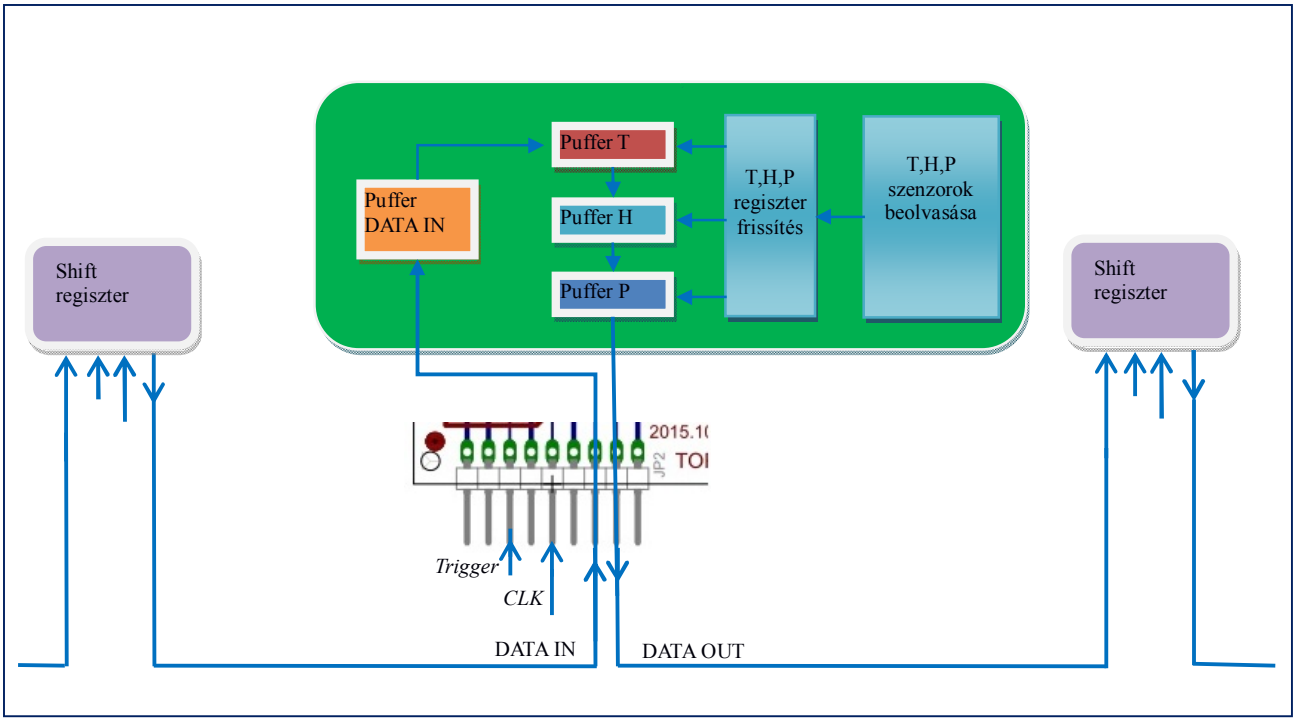
A gyártás és beültetés után a 8. ábrán látható a megvalósult kártya, a baloldalon egy beültetett, a jobb oldalon még egy beültetésre váró nyák látható.



8. ábra – Egy beültetett, és egy beültetésre váró nyák

3.7. Programterv

A mikrokontroller program a 3.3-as fejezetben ismertetett elvek mentén kell, hogy működjön. Azaz szimulálnia kell, az adatbuszon történő olvasáskor, mintha a három szenzor értékei sorba kötött shiftregiszterekben tárolódnának.



9. ábra - Mikrokontroller programfelépítés vázlat

A 9. ábrában a zöld háttérrel jelölt részbe rajzoltam le a mikrokontrollerben futó program vázlatos viselkedését. A szenzoroktól adott időközönként beolvassuk a mért értékeket a mikrokontrollerbe I^2C-n keresztül. A trigger jel jelzi, hogy adat küldés indul, ennek hatására egy megszakításban frissítjük a szenzorok legutóbb beolvasott értékeire a Puffer T, Puffer H és Puffer P változókat. (A kettős pufferelésre azért van szükség, hogy a szenzor értékek frissítése ne írja felül a küldő puffereket adat küldés közben.) A CLK órajelre szinkronizálva, megszakításban a DATA IN lábon érkező bitet beolvassuk, miközben a DATA OUT lábra kitesszük a nyomás puffer (P) utolsó bitjét. A következő órajelek hatására sorra léptetjük be a beolvasott biteket és léptetjük ki a szenzor pufferek (THP) bitjeit. Ha minden szenzor bit kilépett, akkor jön sorra a DATA IN lábon beléptetett adat bitek kiléptetése.

4. A müontomográf GSM/GPRS kapcsolatának kiépítése

4.1. Raspberry Pi

A müontomográf rendszerének fő eleme a Raspberry Pi (röviden RPi) mini számítógép. Erre a számítógépre érkeznek be a shiftregiszterek kiolvasott jelei, az RPi felelős a megkapott mérési adatok feldolgozásáért. Az RPi-n operációs rendszer fut, több operációs rendszer közül lehet választani, több programnyelven programozható. Jelen projektben a Raspberry Pi 2 model B –t használjuk. Műszaki paramétereit az alábbi pontokban részletezem [8]:

- A 900MHz quad-core ARM Cortex-A7 CPU,
- 1GB RAM,
- 4 USB port,
- 40 GPIO pin,
- Full HDMI port,
- Ethernet port,
- Kombinált 3.5mm audio jack és video kimenet,
- Camera interfész (CSI),
- Display interfész (DSI),
- Micro SD kártya hely,
- VideoCore IV 3D grafikus mag.

A modulon elérhető operációs rendszerek az alábbiak lehetnek [9]:

- Raspbian (Hivatalos),
- Windows 10 IOT (Third party),
- Ubuntu Mate (Third party),
- RISC OS (Third party),
- OSMC (Third party),
- stb.

A GPIO lábakon elérhető több kommunikációs interfész is, ilyenek lehetnek például az UART, I2C, SPI. Emellett találhatóak még 5V, 3,3V és GND lábak is.

Ebből azt emelném ki, hogy a GPIO lábakkal, azok funkcióival és a kommunikációs

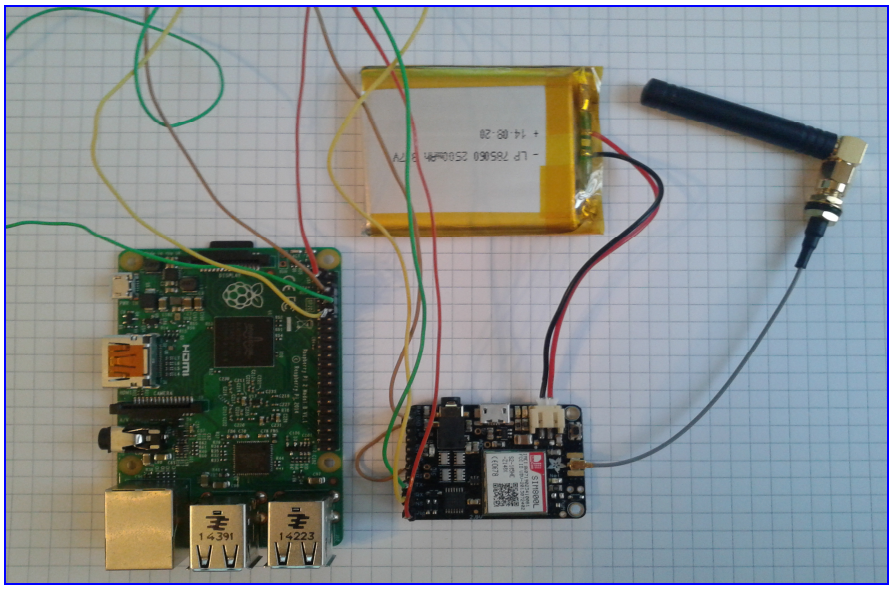
lehetőségeivel többet nyújt, mint egy átlagos PC. Ezek használatával könnyen lehet kapcsolódni más áramkörökhöz vagy programozni vezérléseket. Ezen felül számos kiegészítő termék és áramkör kapható hozzá.

Felmerül a kérdés, hogy miért ezt az eszközt használják a detektornál, miért nem valami cél áramkört alkalmaznak, hiszen sokkal többre képes az RPi annál, mint ami adott esetben feltétlenül szükséges. A válasz a rugalmasságában és a relatíve alacsony árban keresendő. A kutatási csoportok fő építőkövei fizikusok, akik általában több detektort párhuzamosan fejlesztenek. Jelentős időt, pénzt és energiát spórolnak meg azon, hogy nem kell minden detektornál a központi feldolgozó egységet egyedileg megtervezni, legyártatni, felprogramozni és tesztelni. Estelegesen egy nem várt hiba miatt az egész folyamatot megismételni.

A karbantarthatóságot is figyelembe kell venni. Sokszor különböző területekről érkező emberek dolgoznak egy-egy projekten hosszabb-rövidebb ideig, akiknek a munkáját átveszi a kutatócsoport. Ekkor nem árt, ha közös platformon, közös programnyelven fejleszt a csapat, és csak a feltétlenül szükséges részekre fejlesztenek célhardvert.

4.2. GSM/GPRS modul

Az RPi mellé kerestem egy olyan kisméretű GSM/GPRS modult melynek egyszerű a használata és viszonylag olcsó. Választásom az Adafruit cég FONA termékére esett. A modult AT parancsokkal lehet vezérelni UART-on keresztül. A modul lelke a SIMCom által gyártott SIM800 GMS modul. A 10. ábrán láthatunk egy Raspberry Pi PC-t és egy hozzá csatlakoztatott GMS/GPRS modult.



10. ábra - Raspberry Pi és a GSM modul

Pontokba szedve ismertetném főbb paramétereit [10]:

- négy sávos 850/900/1800/1900MHz,
- képes hang hívás küldésére és fogadására,
- képes SMS küldésére és fogadására,
- képes GPRS adat küldésére és fogadására (beépített protokoll támogatás: TCP/IP, http stb.),
- rádióállomásokat képes keresni és fogi,
- AT parancs támogatás,
- UART interfész,
- csatlakoztatható külső akkumulátor,
- az akkumulátor töltéséhez mini USB csatlakozó (max 500 mA-rel),
- 8 Ohm-os hangszóró kimenet,
- 3.5 mm headset csatlakozó.

A feladatra a modul uFL verzióját választottam, ahol az antenna egy adapter kábellel csatlakozik. Kezdetben úgy véltem, hogy a mérési eredmények továbbításához - mivel nem idő kritikus műveletről van szó - elég lesz a GPRS kapcsolati sebesség. Ha esetleg a későbbiekben nagyobb sebesség kell, akkor kaphatóak ebből a termékcsaládból 3G képes verziók is.

4.2.1. Lábkiosztás és LED kijelzések

A modul pin kiosztása a következő volt [10]:

- **Vio** – a modult csatlakoztathatjuk 5V - 3V logikai jelszinten kommunikáló eszközökhöz, ezt a lábat a kommunikációban használt jelszintre kell fixen kötni.
- **Key** – ennek a lábnak a segítségével lehet ki-be kapcsolni a modult. Magas szintről két másodpercig alacsony szintre kell húzni, hogy be- illetve kikapcsoljuk. Most a földre van kötve

így folyamatosan üzemel a tápfeszültség megszűnéséig. Fogyasztáscsökkentés végett később ki lehet alakítani a Burst üzemmódot.

- **PS** – Power Státusz láb, logikai magas szintre megy, ha üzemkész állapotba került a modul.
- **NS –** Network Státusz láb, ami ki van vezetve a Net-ledre is, impulzusokkal jelzi a GSM modul aktuális állapotát.
- **Reset** – 100 ms-ig földre húzva ezt a lábat hardveresen újraindul a modul.
- **Rx** és **Tx** lábak az UART kommunikációhoz.
- **RI** - Ring indicator – hívásjelző láb, alapból magas szinten van, impulzusokkal jelzi, ha hívás érkezik, SMS-érkezésének jelzését is ki lehet vezetni.
- **SPK +/- –** hangszóró csatlakozáshoz.
- **MIC +/- –** mikrofon csatlakozáshoz.
- **BAT** – forraszthatjuk is a külső akkumulátort, ha nem akarjuk a dedikált csatlakozót használni.
- **GND** – föld.
- **PWM –** PWM kimenet.
- **AD –** Analog-Digital átalakító bemenet.

A kijelző LED-ek az alábbiak szerint változhatnak:

- **PWR** – kéken világít, amint a modul használatra kész.
- **NET –** pirosan világít és
  - 64ms – be és 800ms - ki impulzussal jelzi, hogy a modul használatra kész, de még nem kapcsolódott mobilhálózatra vagy
  - 64ms – be és 3s - ki impulzussal jelzi, hogy a modulnak van hálózati kapcsolata és képes hívás, illetve SMS küldés/fogadásra vagy
  - 64ms – be és 3ms - ki impulzussal jelzi, hogy a modulnak van GPRS kapcsolata.
- **Charging** – narancsszínnel világít akkutöltés alatt.

- **Done** – zöld színnel világít, ha az akkumulátor feltöltődött.

4.2.2. Fogyasztás

SIMCom adatlapból [11] tudhatjuk meg a GSM/GPRS modul fogyasztási adatait, amik a következőképpen alakulhatnak (Vbat = 3,8V esetén):

- Kikapcsolt állapotban: 50-60 uA
- Sleep módban: 1.0 – 1,8 mA
- Idle módban: 18.7 mA
- Hívás közben (kapcsolatok típusától függően): 131mA – 216 mA
- GPRS (kapcsolatok számától és típusától függően): 148mA - 453mA
- Imax peak áram Tx induláskor 2A –is elérheti.

4.3. Kapcsolat a Raspberry Pi-vel

Az RPi-hez való csatlakozáshoz a következő lábakat kell összekötni:

1. A GMS modulon a Key lábat a földre kell kötni.
2. GSM modul GND ⟷ RPi GND
3. GSM modul Rx ⟷ RPi Tx
4. GSM modul Tx ⟷ RPi Rx
5. GSM modul Vio ⟷ RPi 5V

Természetesen emellett az akkumulátort csatlakoztatni kell a GSM modulhoz.

4.4. GPRS kapcsolat kiépítése a Raspberry Pi –n

A GMS modulunk gyakorlatilag egy GSM/GPRS soros modem, ami UART interfésszel rendelkezik. Az UART soros kapcsolatot a Raspbian kernel is támogatja, ezt kihasználtam a Raspberry Pi-n, ahol Linux operációs rendszerre már vannak kész megoldások telefon modemes kapcsolat kezelésére. Amire szükség volt az PPP (Point to Point Protocoll) kezelő szoftver, melynek

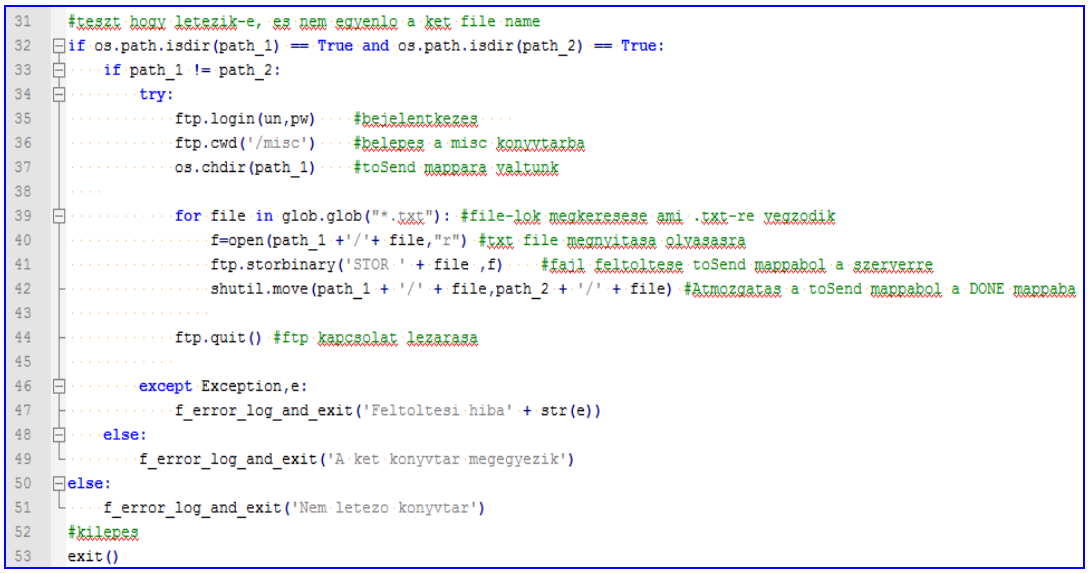
segítségével soros vonalon csatlakozhatunk az internetre. Erre megoldás nyújt a ppp screen elinks szoftver. Ez a szoftver egy másik beépített szolgáltatással dolgozik együtt, aminek a neve chat script. Ez fogja kezelni az AT parancsok küldését és a válaszok értelmezését a PC és a modem között [12][13]. A GSM modul gyártójának „learning center”-ében található egy beüzemelési útmutató ahol leírásra kerül a Raspberry Pi és a Fona termék beüzemelése [14].
Ezt végrehajtva sikeresen felépítettem a soros vonali kapcsolatot, a GMS modul és a Raspberry Pi között, és csatlakoztam a mobilinternet hálózatra.

4.4.1. További lehetőségek

A projekt e fázisában a fő cél a már feldolgozott mérési eredmények eljuttatása egy szerverre. A későbbiek folyamán a GSM modul alkalmazásával lehetőség nyílik esetleges üzemzavarok, meghibásodások jelzésére SMS-ben, vagy esetleges karbantartási szükségletek jelzésére. Terepi eszköz lévén ez nagyon hasznos funkció lehet, hiszen így a karbantartónak csak akkor kell a helyszínre mennie, ha feltétlenül szükséges.

4.5. Mérési adatok küldése távoli szerverhez

Ha már van felépült internet kapcsolatunk az RPi-n, akkor már csak egy szervernek kell eljuttatni a mérési eredményeket. A mérési eredményeket egy másik fél által írt program szolgáltatja. Ez még egyeztetés alatt van, de lényegében text fájlokról beszélünk melyek egy „toSend” küldési mappába kerülnek. Ezeknek a text fájloknak két típusa van, az egyikben a hőmérséklet, nyomás, páratartalom, hatásfok adatok vannak, a másikban koordináták szerint a sűrűség eloszlás.



```
#teszt hogy letezik-e, es nem egyenlo a ket file name
if os.path.isdir(path_1) == True and os.path.isdir(path_2) == True:
    if path_1 != path_2:
        try:
            ftp.login(un,pw)    #bejelentkezes
            ftp.cwd('/misc')    #belepes a misc konyvtarba
            os.chdir(path_1)    #toSend mappara valtunk

            for file in glob.glob("*.txt"): #file-lok megkeresese ami .txt-re vegzodik
                f=open(path_1 +'/'+ file,"r") #txt file megnyitasa olvasasra
                ftp.storbinary('STOR ' + file ,f)    #fajl feltoltese toSend mappabol a szerverre
                shutil.move(path_1 + '/' + file,path_2 + '/' + file) #Atmozgatas a toSend mappabol a DONE mappaba

            ftp.quit() #ftp kapcsolat lezarasa

        except Exception,e:
            f_error_log_and_exit('Feltoltesi hiba' + str(e))
    else:
        f_error_log_and_exit('A ket konyvtar megegyezik')
else:
    f_error_log_and_exit('Nem letezo konyvtar')
#kilepes
exit()
```

11. ábra - FTP kezelő program

Írtam Python nyelven egy programot, aminek a lényegi része a 11. ábrán látható. Ennek a szkriptnek két bemenő paramétere van (**path_1, path_2**). Az első a „**toSend**” mappa teljes elérési útja, amiben azokat a text fájlokat tartalmazza, amelyeket el szeretnénk küldeni a szerver felé, a második paraméter az a „**Done**” mappa teljes elérési útja, ahova át szeretnénk helyezni a már elküldött text fájlokat. A textek elküldésére a szerver felé FTP protokollt használok. Ha valami hiba történik, akkor a szkriptet tartalmazó mappába létrehozok egy **error_log.txt** fájlt, és beleírom a hibát, majd kilépek a programból. Ezt a szkriptet lehet akár ütemezett feladatként is futtatni, vagy beépíthető a kiértékelő algoritmus végébe, ez még megbeszélés tárgya.

5. Szerveroldali programozás

5.1. Webszerver applikáció fejlesztése XAMPP

Olyan nyelveket kerestem, melyek nyílt forráskódúak, fontos tényezők voltak még a költséghatékonyság, megbízhatóság, és hogy jól dokumentált nyelvek legyenek, amit önállóan el tudok sajátítani. Ezekből kifolyólag döntöttem egy Linux alapú webszerver, a PHP, a MySQL és a kliens oldali JavaScript és HTML nyelvek mellet.

A webszerver programozás során, a szerver oldalon (back end) PHP-t és adatbázisnak MySQL-t használok, míg a megjelenítésre (front end) HTML-t és JavaScript-et. Arra kerestem egy programot, hogy a fejlesztéseket a saját PC-en tudjam elvégezni. Erre nyújtott megoldást a XAMPP nevű program, ami egy lokális gépen használható, nyílt forráskódú, platform független, Linux - Apache webszerver disztribúció. A telepítése és konfigurálása igen egyszerű, és minden szükséges szoftverkomponenst és értelmezőt tartalmaz. A XAMPP szó összetevői: X- cross angol szóra utal, A- Apache webszerver, M – MySQL, P – PHP, P – perl [16][17].

A fejlesztéshez használt mappákat és fájlokat a ..\xampp\htdoc\ mappába kell tenni. A konfiugrátorban el kell indítani az Apache szervert és az adatbázis-kezelőt, majd a webböngészőbe beírva a localhost/<file név> -en keresztül jeleníthető meg az oldalunk. Több szolgáltatást is el lehet érni a XAMPP-ból, ha annyit írunk be a böngészőbe, hogy localhost. Ez azért hasznos, mert ilyenkor betöltődik a PHPadmin felület, amin keresztül elérhetjük grafikus felületen a MySQL adatbázist. Itt tesztelhetjük a lekérdezéseket, hozhatunk létre, módosíthatunk, törölhetünk adatbázisokat. A fejlesztéshez Komodo ingyenes fejlesztőkörnyezetet használtam, bár a highlight-on kívül nem sok szolgáltatása van, egyszerű szövegszerkesztőként működik.

5.2. Az alkalmazott programnyelvek és könyvtárak rövid bemutatása

Ebben a fejezetben nagyon röviden sorra venném a használt programnyelveket és könyvtárakat. Ezekről a nyelvekről és könyvtárakról sok irodalom született már, teljes körű megismerésük éveket vesz igénybe. Tehát ebben a fejezetben csak egy rövid, lényegre törő képet szeretnék festeni a programnyelvekről, így hozzásegítve a diplomamunkám olvasóját a programom megértéséhez.

5.2.1.HTML

Weboldal készítésekor a kliens oldali megjelenítéshez szükség van egy olyan nyelvre amely „megmondja” a böngészőknek, hogy a betöltött weblapot, hogyan jelenítse meg. Erre való a HTML (HyperText Markup Language), ami a jelölő nyelvekhez tartozik. A HTML önmagában nem sok mindenre képes, a weboldal struktúráját határozza meg. Hogy vizuálisan szép legyen, ki lehet egészíteni CSS-sel (Cascading Style Sheets), ami kifejezetten a megjelenítéshez ad plusz információt. Hozzá lehet adni logikákat, függvényeket, amik a kliens oldalon futnak JavaScript nyelven, vagy szerver oldalról is kérhetünk, kaphatunk megjeleníthető HTML kódot [18][19].

A HTML fájlok, szöveges fájlok, amiknek meghatározott felépítésük van: típus definíció, fejléc, és törzs. Minden elemnek van egy nyitó és egy záró tag-je. Négy fajta szimbólum csoportból áll [18][19]:

1. Strukturális: az adott elem hol helyezkedjen el, pl. címsor
2. Prezentációs: Dőltbetű, félkövér - ez mára elavult, CSS-t használnak helyette
3. Hivatkozások, linkek
4. Eszköz elemek: gombok, listák, beviteli mezők,…stb.

Ezen kívül lehetőség van az oldalon JavaScript használatára. A <script> </script> tag-ek között valósíthatjuk meg a függvényeket, behúzhatunk külső könyvtárakat.

Egy HMTL elemhez rendelhetünk eseményeket, például „onclick” eseményt, melynek hatására hívhat egy JavaScript függvényt. Ez a függvény képes kiolvasni ID alapján bármilyen HTML elemből, hogy mi az az aktuális státusza (például egy legördülő listából melyik érték van kiválasztva) és ID alapján el tud érni HTML elemeket és azoknak tud átadni adatot (pl. egy HTML gomb elemre történő kattintás meghív egy JavaScript függvényt és az feltölt egy másik HTML elemet tartalommal.). Egy ilyen esemény hozzárendelést láthatunk a 12. ábrán, a 13. ábrán pedig láthatjuk, hogyan kell megvalósítani JavaScriptben a HTML elemek állapotának kiolvasását.

```
552         </select>
553         <input id="START_FLUX_MEGHIVAS" type="button" value="Show" onclick="START_FLUX()"
554 <!-- MEGJELENÍTÉS rész -->
```

12. ábra - Függvényhívás hozzárendelés HTML gombhoz

```
197     //FLUX kirajzolás
198     function START_FLUX(){
199
200         var JS_Time_from = $('#select_FLUX_FROM').val()
201         var JS_Time_to = $('#select_FLUX_TO').val()
202         var JS_SelectedRPI = $('#selectrpi_FLUX').val()
203
```

13. ábra - HTML elemek állapotának kiolvasása JavaScripttel

A JavaScript kódon kívül elhelyezhetünk PHP kódot is a HTML sorok között, ezt a <?php ?> tag közé kell illeszteni, ahogyan a 14. ábra is szemlélteti. Ezek a sorok szerver oldalon fognak műveletet végezni, majd visszatérhetnek adott esetben HTML kóddal is, amit a fenti példákhoz hasonlóan szintén meg tudunk jeleníteni a böngészőben. Egy példa látható PHP kód beszúrására a 15. ábrán.

```
531         <select id="selectrpi_FLUX" onchange="Fill_Rpi_FLUX_Dates()">
532             <option value="" disabled selected>Select RPi</option>
533         <?php
534             include_once('..\WebServ\DBHandrel.php');
535
536             $sql = "SELECT RPi_name FROM flux_lookup group by RPi_name ORDER BY RPi_name";
537
538             $response = f_mySql_query($sql);
539
540             while($row = mysqli_fetch_array($response)){
541                 echo "</br>" . $row['RPi_name'] ;
542                 echo "<option value=\"". $row['RPi_name'] ."\">" . $row['RPi_name'] . "</option>";
543             }
544         ?>
545             <option value="A_selected">Aggregated</option>
546         </select>  from
```

15. ábra - PHP kód beszúrása HTML kódba

Nemrég jelent meg a HTML legújabb verziója a HTML5, ahol lényegében megőrizték az eredeti felépítést és a visszafelé kompatibilitást, de kibővítették a funkcionalitását a nyelvnek és új tag-eket vezettek be. A fő céljuk az volt, hogy a plug-in-ekre (pl. Adobe Flash, Silverlight) többé ne legyen szükség [18][19].

Egy kis érdekesség, hogy a HTML nyelv kitalálója egy angol fizikus, kutató Sir Timothy John Berners-Lee nevéhez fűződik, aki 1990-es években a CERN-ben dolgozott, miközben feltalálta a WWW-t, megírta az első böngészőt, ami az általa kifejlesztett HTML szöveges fájlokat tudta értelmezni [20].

5.2.2. MySQL

A MySQL egy relációs adatbázis-kezelő szerver. Van fizetős és ingyenes változata, jelenleg

az Oracle tulajdonolja, eredetileg egy svéd cég fejlesztette ki C/C++ nyelven 1995-ben. A világ vezető adatbázis kezelő szoftverét olyan nagy cégek is használják, mint a Google vagy a Facebook, de az egészen kis weblapok körében is előszeretettel használják [21]. Együtt alkalmazva a PHP, JavaScript és HTML nyelvekkel, dinamikus tartalmú weblapokat készíthetünk. Az alkalmazott adatbázis felépítését részletesen bemutatom a 5.5 MySQL adatbázis felépítése című fejezetben.

5.2.3. PHP

A nyelvet Rasmus Lerdorf 1994-ben fejlesztette ki, a PHP eredetileg a Personal Home Page Tools szavak kezdőbetűiből alakult ki, manapság már a Hypertext Preprocessor-ból származtatják. A PHP egy nyílt forráskódú szkriptnyelv, ami a szerver oldalon fut. Segítségével dinamikus honlapokat készíthetünk. A szerver oldalon szükség van egy PHP fordító modulra. Az alapvető működése az, hogy értelmezi a kliens oldalról érkező kéréseket, műveleteket végez, majd HTML kód formájában válaszol a kérésekre. Mivel szerver oldalon fut, ezért a kód nem elérhető a kliens oldalon.[22] Az általam írt PHP szkripteketről a 5.6 fejezetekben írok.

5.2.4. JavaScript

A JavaScript (JS) egy magas szintű, objektum orientált, típus nélküli nyelv, melyet főleg webes alkalmazások készítéséhez használnak. Szintakszisa hasonló a Java nyelvhez. Kliens oldalon végezhetünk vele művelteket, a kód letöltődik a kliens oldalra. Első változatát Brendan Eich 1995-ben fejlesztette ki, mikor még a Netscape-nél dolgozott.[23] Az általam írt JavaScript kódról a 0 fejezetben írok részletesen.

5.2.5. Adat átadás a PHP és a JavaScript között: JQery, Ajax, JSon

A JQuery egy JavaScriptnek egy nyílt forráskódú könyvtára, ami azért jött létre, hogy segítse a kliens oldali HTML elemekkel való kapcsolatot. Ennek a könyvtárnak egy része az Ajax támogatás, ami pedig arra szolgál, hogy kliens - szerver lekérdezés - válasz esetén ne keljen az egész oldalt újra betölteni. [25][24] A JSon pedig egy általános szöveges tömb formátum. Ez arra szolgál, hogy olyan struktúrában adjuk át az adatot a PHP és a JavaScript között, amit mindkét fél fel tud dolgozni. A gyakorlatban ez annyit tesz, hogy a PHP oldalon kiszámolt eredményeket az átadás előtt JSon formátumra konvertáljuk, amit aztán a JavaScript oldalon át tudunk alakítani

tömbbé.

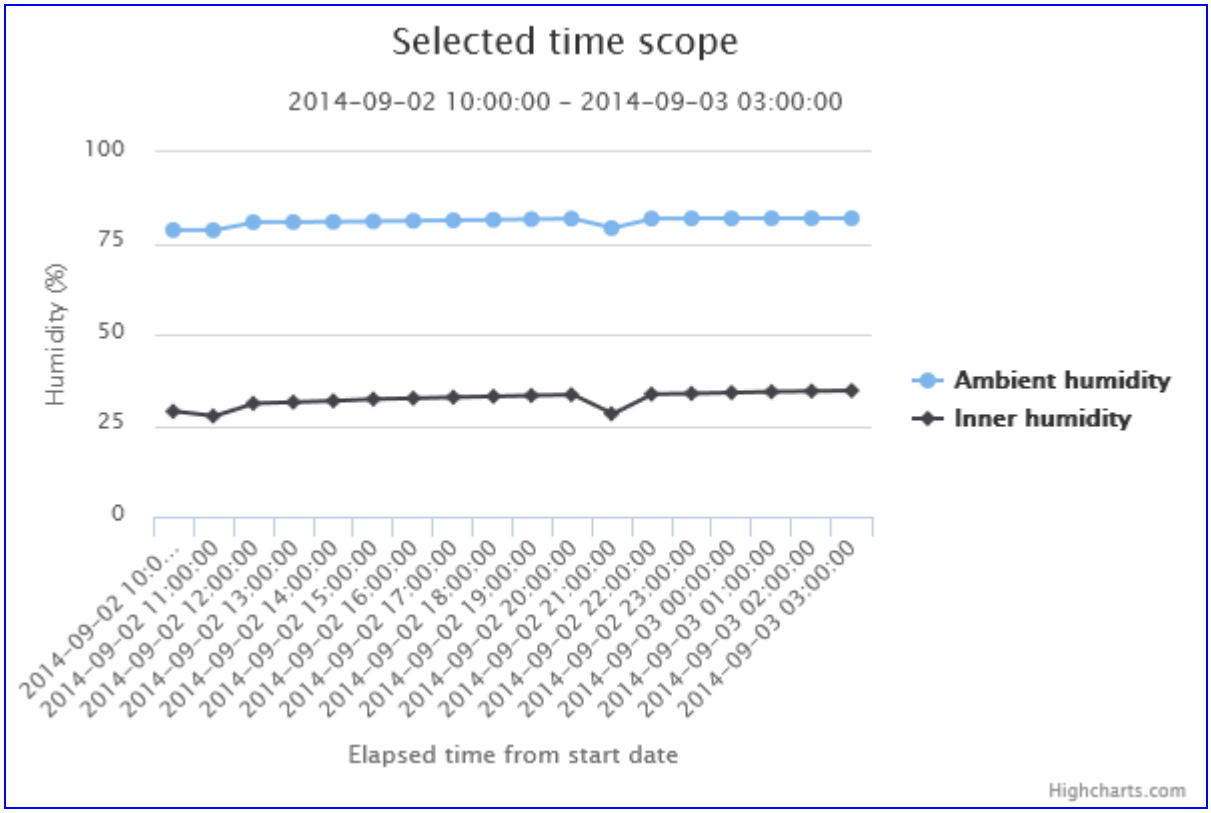
```
238     if (JS_SelectedRPI != 'A_selected') {
239         //Akkor rpi-t választott
240         var JQueryAjax = $.post( 'QueryHandler.php?TYPE='+5+'&RPI='+JS_SelectedRPI+'&DATE_FROM='+JS_Time_from+'&DATE_TO='+JS_Time_to+'&SENSOR='+0)
241         .done(function(data) {
242                 data = JSON.parse(data)
```

16. ábra - JQueryAjax hívás

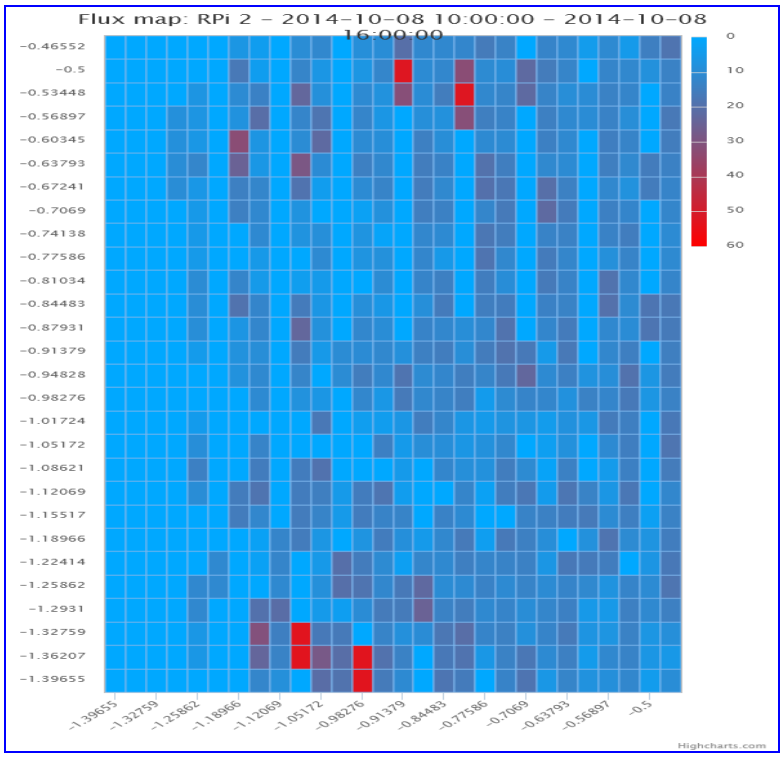
Látható a 17. ábrán, hogy a JQuery belsejében egy PHP szkript fájlt a QueryHandler.php-t hívom meg. A szkript bemenő paramétereit „?” jel után kell megadni. A „.done” metódusa akkor hívódik meg, amikor a PHP szkript befejezte a futását, az így visszakapott adatokat a „data” paraméter tárolja, jelen esetben JSon formátumban, melyet feldolgozva, használhatunk tovább a JavaScript oldalon.

5.2.6.Megjelenítés – HighChars

A grafikonok, chartok rajzolásához a Highsoft AS - egy norvég cég - által fejlesztett JavaScript könyvtárat használom, melynek a neve HighCharts. A program nem kereskedelmi célra ingyenesen használható. Rengeteg diagram típus közül lehet választani, mitöbb a weboldalukon [26] kipróbálható az összes demó chart. Itt lehetőségünk nyílik megtekinteni a forráskódot, amit szerkeszteni és futtatni is tudunk a demó felületen. Sok beépített funkció közül választhatunk a program használata során: ilyen például a tooltip lehetőség, ami a grafikon pontjain jeleníti meg a konkrét értékeket. Továbbá választhatjuk az automatikus felirat elrejtést is, ha nagyon kis távolságon sok feliratot kellene megjelenítenünk, de hely hiányában nem áll módunkban ezt megtenni. Arra is van lehetőség, hogy a program automatikus label forgatást, más néven feliratforgatást hajtson végre akkor, ha túl hosszú lenne a megjelenítendő szöveg. Ez a program számos formátumban (jpeg, png, pdf, svg) kínálja letöltésre és nyomtatásra a grafikonokat. A 18. és 19. ábrán egy-egy HighCart-tal rajzolt, THPE és fluxus diagram látható.



20. ábra - Relatív páratartalom megjelenítése az idő függvényében, HighChart könyvtár használatával



21. ábra - Fluxus eloszlás megjelenítése, HighChart könyvtár használatával

Ahogy a 21. ábra is látszik nem teljesen tökéletes az exportálás heat map esetében, mert a heat map cím mezője belelóg a képen az értékes mezőkbe, holott a honlapon még nem lógott bele.

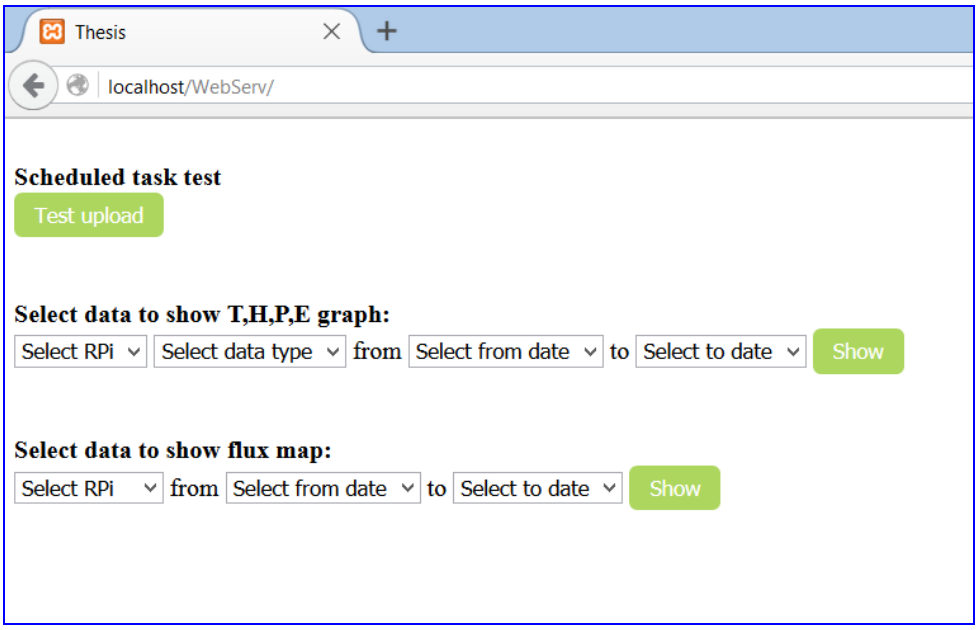
5.3. A webszerver mappa és fájlrendszerének a felépítése

A webszerveren található mappák, fájlok, és azok funkciójának rövid leírása az alábbiakban olvasható: (Ha külön nincs jelölve, minden fájl és könyvtár a ..\xampp\htdoc\WebServ könyvtárban helyezkedik el.)

- **INCOMING** mappa: ebbe a mappába töltik fel az RPi-k a mérési eredményeiket tartalmazó fájlokat,
- **DONE** mappa: ebbe a mappába kerülnek azok a fájlok, amik sikeresen bekerültek az adatbázisba,
- **ERROR** mappa: ide kerülnek a hibás fájlok, amiket nem lehetett valamiért feltölteni az adatbázisba, szintén itt található még egy hiba bejegyzés fáj,
- ..\xampp\htdoc\WebServ\ERROR**error_log.txt** fájl: adatbázis művelet hiba esetén, hiba bejegyzés készül, mely ebbe a fájlba íródik,
- **DBHandler.php** fájl: az adatbázishoz való kapcsolódás, lekérdezések átadása, és válaszok konvertálása a feladata,
- **IncomHandler.php** fájl: az RPi-ktől beérkezett fájlokat feltölti az adatbázisba,
- **PlotChart.php** fájl: a megjeleníteni kívánt Chart típust betölti a weblapra,
- **QueryHandler.php** fájl: a weblap felől érkező információk alapján összerakja és végrehajtja a megfelelő lekérdezéseket,
- **index.php** fájl: ez tartalmazza a kliens oldalon megjelenő weblapot, a HTML és JavaScript kódokat
- **jquery.js** fájl: Jquery JavaScript könyvtár,
- **json2.js** fájl: JSon formátumkonvertáló JavaScript könyvtár

5.4. A Weblap felépítése

5.4.1. A Weboldal megjelenítése böngészőben



22. ábra - A weboldal felépítése

A weboldal címének beírásakor a böngésző betölti az index.php fájlt a kliens oldalra. Ez a fájl HTML szöveges állományt tartalmaz kiegészítve JavaScript és PHP kóddal. Struktúráját tekintve, a HTML törzs felső részbe helyeztem el minden JavaScript könyvtárhivatkozást és kódot, alatta pedig a HTML kód található, mely tartalmaz beágyazott PHP kódot is. A megjelent weboldal a 22. ábrán látható. A weboldalon a felhasználó THPE (Temperature, Humidity, Pressure, Efficiency) megjelenítés esetén rögtön ki tudja választani, hogy melyik RPi adatait szeretné megjeleníteni. Ha ezt kiválasztotta, akkor tud dátumot választani, hogy mely időintervallum közötti adatokra kíváncsi. Ettől függetlenül beállíthatja, hogy hőmérséklet, páratartalom, nyomás vagy hatásfok ábrát szeretne, mely, ha minden adatot beállított, a THPE sorban lévő Show gombbal jelenik meg. Egy ilyen lekérdezés eredménye a 23. ábrán látható. A fluxus eloszlás megjelenítése hasonlóan történik.

23. ábra - THPE megjelenítés böngészőben, adott időintervallumra

5.4.2. A weboldal HTML elemei

A weboldalon használt HTML elemek leírása a 24. összefoglaló táblázatban olvashatók. Ebben a táblázatban megtalálható az adott HTML elem látható neve, típusa, ID-ja, hozzárendelt eseménye, hogy milyen függvényt hív és funkciójának rövid ismertetése (a szöveg elemek kivételével):

HTML elem látható neve	Típus	ID	Esemény	Hívott függvény	Funkció
Test upload	button	TEST_UPLOAD_BTN	onclick	testUploadBtn	php szkript indítása, ami a feltölti a mérési adatokat az adatbázisba (kizárólag teszt)
Select RPi (THPE)	select	selectrpi_THPE	onchange	fillRpiTHPEDates	RPi kiválasztó

					gomb
Select data type	select	selectTHPE	-	-	T,H,P,E lekérni kívánt adattípus választó
Select from date	select	select_THPE_FROM	-	-	Adott Rpi-hez tartozó kezdő dátumválasztó
Select to date	select	select_THPE_TO	-	-	Adott Rpi-hez tartozó vég dátumválasztó
Show (THPE)	button	SHOW_THPE_CALL_BTN	onclick	showTHPEChart	A kiválasztott T,H,P,E chart megjelenítése
Select RPi (FLUX)	select	selectrpi_FLUX	onchange	fillRpiFluxDates	RPi kiválasztó gomb
Select from date	select	select_FLUX_FROM	-	-	Adott Rpi-hez tartozó kezdő dátumválasztó
Select to date	select	select_FLUX_TO	-	-	Adott Rpi-hez tartozó vég dátumválasztó
Show (FLUX)	button	SHOW_FLUX_CALL_BTN	onclick	showFluxHeatMap	A kiválasztott RPi vagy összesített fluxus eloszlás megjelenítése
-	div	THPE_MEGJENEITES	-	-	Ebbe az elembe töltődik a THPE chart
-	div	FLUX_MEGJENEITES	-	-	Ebbe az elembe töltődik a Flux heat map
-	div	TEST_MEGJENEITES	-	-	Ebbe az elembe töltődnek a debug üzenetek, amit az IncomHandler.php

					küld

24. táblázat - Az alkalmazott HTML elemek tulajdonságai

Külön meg kell említenem, hogy a selectrpi_THPE és selectrpi_FLUX ID –val jelölt elemek beágyazott php kóddal rendelkeznek. Ezek a kódok a betöltéskor aktivizálódnak, egy lekérdezést indítanak az adatbázis felé, melyben lekérik az összes eddigi RPi nevét ami THPE vagy Fluxus adatot küldött. A választ ezután beillesztik a HTML kódba, és már ez jelenik meg a kliens oldalon. Ez azért volt hasznos, mert a betöltés után ki tudja választani a felhasználó, hogy melyik RPi-től szeretne adatot megjeleníteni.

5.4.3. A kliensoldali JavaScript függvények leírása

Az alábbi felsorolásban találhatóak az általam írt JavaScript függvények nevei és funkciójának rövid leírása:

- **testUploadBtn**: Ez a függvény lefuttatja az IncomHandler.php fájlt, mely feltölti az adatbázisba az RPi-ktől érkezett mérési eredményeket az adatbázisba. Ennek a fájlnak a futtatása ütemezett feladat lesz a szerveren, csak tesztelési célokat szolgál ez a függvény.
- **fillRpiTHPEDates**: Ez a függvény a QueryHandler.php szkriptet hívja, lekérdezi a kiválasztott RPi-hez tartozó összes dátumot, amikor THPE adatot küldött. Majd feltölti a dátumokat a dátum kiválasztó HTML elemekbe.
- **fillRpiFluxDates**: Ez a függvény a QueryHandler.php szkriptet hívja, lekérdezi a kiválasztott RPi-hez tartozó összes dátumot, amikor fluxus eloszlás adatot küldött. Majd feltölti a dátumokat a dátum kiválasztó HTML elemekbe.
- **showFluxHeatMap**: A kiválasztott RPi-t, és a kiválasztott dátumokat átadja QueryHandler.php-nek és három tömböt vár vissza az x,y,z értékek tömbjeit , amit aztán a HighChart-nak feldolgozható alakra formázás után, átad a **drawFluxHeatMap** függvénynek.
- **showTHPEChart**: A kiválasztott RPi-t és a kiválasztott dátumokat átadja QueryHandler.php-nek és három tömböt vár vissza (belső, külső szenzorértékeket és a hozzájuk tartozó mérési dátumokat). amit aztán a HighChart-nak elfogadható alakra formázás után, átad a **drawTHPEChart** függvénynek.
- **drawTHPEChart**: A paraméterként átadott adatok alapján kirajzolja a THPE diagramot.

- **drawFluxHeatMap**: A paraméterként átadott adatok alapján kirajzolja a Fluxus eloszlás heat map diagramot.

Látható, hogy minden JavaScript függvény a QueryHandler.php szkriptet hívja, a különbség a bemenő paraméterek között van. Az egyik fontos paramétere a QueryHandler.php-nek a TYPE, ami adott függvényre egy egyedi szám. Ez alapján különbözteti meg a QueryHandler.php, hogy melyik függvény hívta meg, és ez alapján tudja, hogy milyen lekérdezést és választ várnak tőle.

5.5. MySQL adatbázis felépítése

Az adatbázist úgy alakítottam, ki, hogy a lehető legjobban illeszkedjen a feladathoz. Célom első sorban a hőmérséklet, páratartalom, nyomás, hatásfok ábrázolása az időben, különböző RPi-k esetében. Ez egy viszonylag könnyen kezelhető adatmennyiséget jelent, hiszen ezeket a mennyiségeket viszonylag ritkán fogják küldeni. További célom volt az is, hogy a fluxus eloszlást ábrázoljuk adott időintervallumra, integrálva, adott eszköznél. Ez annyit jelent, hogy adott időintervallumra minden x,y koordinátapárra összegezni kell a z koordinátát, ami elég idő és memóriaigényes lehet, tekintve, hogy csak egy mérési adatfájlban akár 25000 x,y,z koordináta bejegyzés is lehet, ezzel számolnom kellett az adatbázis struktúra kialakításakor.

A XAMPP programot elindítva, böngészőnkbe beírva a localhostot elérhető a phpMyAdmin felület, melyen keresztül grafikusan elérhető a MySQL webes, grafikus felülete. Itt létrehoztam egy új adatbázist measurements_db néven, hoztam létre az összes táblát amit használtam a munkám során.

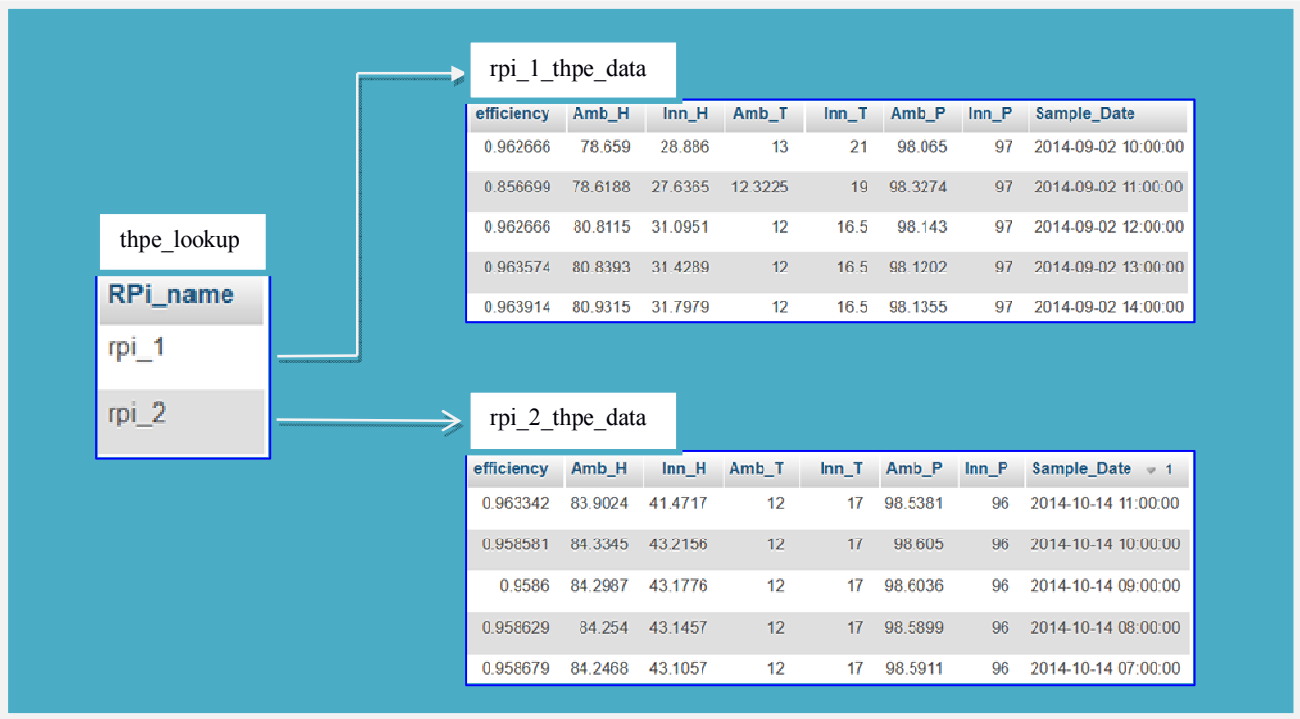
5.5.1. Teszt fájlok feldolgozása

Két különálló mérési fájlt fognak felküldeni az RPi-k. Az elsőt elneveztem THPE típusnak, mert a mért külső és belső hőmérséklet, páratartalom, nyomás valamit hatásfok adat tartalmazza. A második típus Flux típusnak neveztem el, mert az ilyen típusú fájlokban csak x,y,z értékpárok szerepelnek.

Tesztelési céllal kaptam mérési adatokat még egy előző mérésből, ami egy földalatti üregben üzemelő tomográftól származik, ezek között volt 150 darab THPE és 150 darab Flux típusú fájl. A THPE típusú fájlokba beleírta megy plusz nyomás adatot minden méréshez, hogy a jövőbeli mérések struktúrája szerinti adatokkal tudjak dolgozni. Hozzá kell tennem, hogy a Flux fájlok elég nagyméretűek, fájlonként kb. 25000 x,y,z koordináta adatot tartalmaznak.

A teszt adatbázis feltöltésénél a THPE és Flux típusú fájlokat két részre szedtem, mintha két RPi-től származtak volna a fájlok, 100 darab Fluxus és 100 darab THPE fájl az RPi 1 től és 50 darab Fluxus és 50 darab THPE fájl az RPi 2 től.

5.5.2. Adott RPi-hez tartozó hőmérséklet, páratartalom, nyomás, hatásfok és időbélyeg adatok tárolása



25. ábra – RPi, T, H, P, E, dátum adatok tárolása MySQL-ben

Az ötletem az volt, ahogy az a 25. ábrán is látható, hogy legyen egy tábla, amelyikbe azt jegyezzük be, hogy mely RPi-től kaptunk eddig THPE adatot. Ez a thpe_lookup tábla, az itt szereplő RPi nevekhez hozzáfűzve a „_thpe_data” karakterláncot, megtalálhatjuk az adott RPi-hez tartozó táblát az adatbázisban, amiben az RPi mérési adatait és annak időbélyegeit rögzítettem.

A thpe_lookup tábla, csak egy oszlopot tartalmaz, az RPi_name néven, aminek a típusa varchar(10), ez egyben a kulcsként van beállítva, ami minden sorra egyedi kell, hogy legyen, ezért minden RPi név csak egyszer szerepelhet benne. Ez azért fontos, mert a táblaneveket az RPi nevekből képezzük egy fix karakter hozzáfűzésével, ezért az egyértelmű leképezéshez nem

szerepelhet duplán RPi név a táblában.

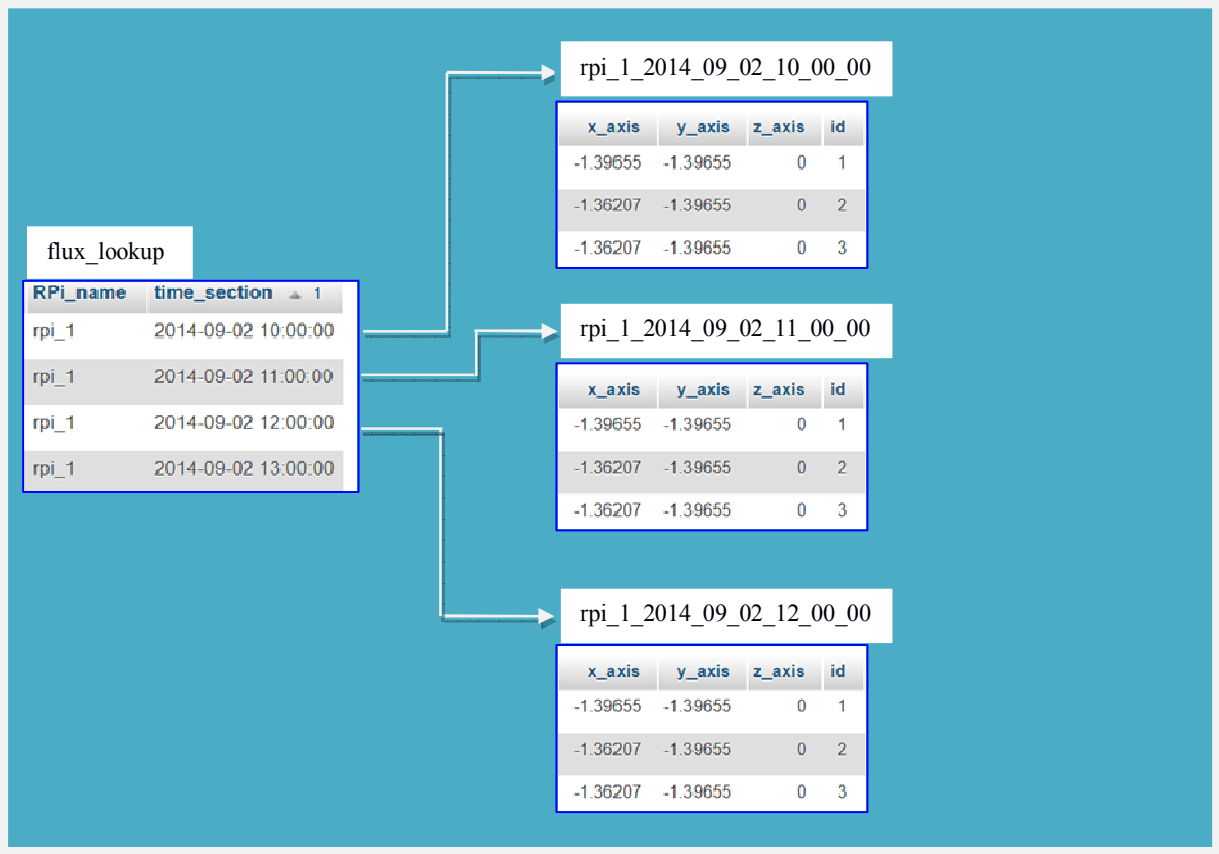
Egy rpi_x_thpe_data table estében kulcsként a Sample_Date van megadva, így nem lehet, kétszer ugyanazzal az időbélyeggel bejegyezni mérési eredményt egy adott RPi esetében.

Az itt szereplő oszlopok neve (típusa) – tárolt jellemző magyarul:

- efficiency (float) - hatásfok
- Amb_H(float) – Külső relatív páratartalom
- Inn_H(float) – Belső relatív páratartalom
- Amb_T(float) – Külső hőmérséklet
- Inn_T(float) – Belső hőmérséklet
- Amb_P(float) – Külső nyomás
- Inn_P(float) – Belső nyomás
- Sample_Date(datetime) – A mérés időbélyege (kulcs)

Mindegyik helyen kell, hogy legyen adat, ezért beállítottam az összes oszlopra egy további jellemzőt a not null-t, ami nem engedi, hogy egy sorban az ilyen tulajdonsággal, ellátott oszlopot üresen hagyjuk.

5.5.3. Adott RPi-hez tartozó fluxus adatok és időbélyeg tárolása



26. ábra – Fluxus adatok tárolása MySQL-ben

Az adat fluxus adatok tárolását hasonlóra terveztem, mint a THPE adatok esetében. Az adattáblák, és az eltárolt információk a sematikus ábrája a 27. ábrán látható. A flux_lookup táblában tárolom az egyes RPi-k neveit és azt, hogy mikor küldtek fluxus adatot. A flux_lookup táblának két oszlopa van, az RPi_name ami varchar(10) típusú, és egy time_section oszlop ami datetime típusú. Ezek az értékpárok együttesen alkotnak egy egyedi kulcsot minden sorban. Így nem szerepelhet azonos RPi név-dátum kombináció. Ez azért fontos, mert az RPi nevét és adatküldési dátumának összefűzéséből találhatjuk meg, adott RPi adott időbélyeggel küldött fluxus adat táblázatát. Ebből nem lehet két azonos, mert akkor nem lenne egyértelmű a tábla hozzárendelés. A táblanév képzése a következő: a dátumból minden kötőjelet, szóközt, és kettőspontot alulvonással helyettesítünk, az RPi nevéhez alulvonással hozzáfűzzük az előző karakter láncot és megkapjuk a táblanevet.

Egy általános rpi_x_YYYY_MM_DD_HH_MM_SS tábla szerkezete a következő képen épül fel:

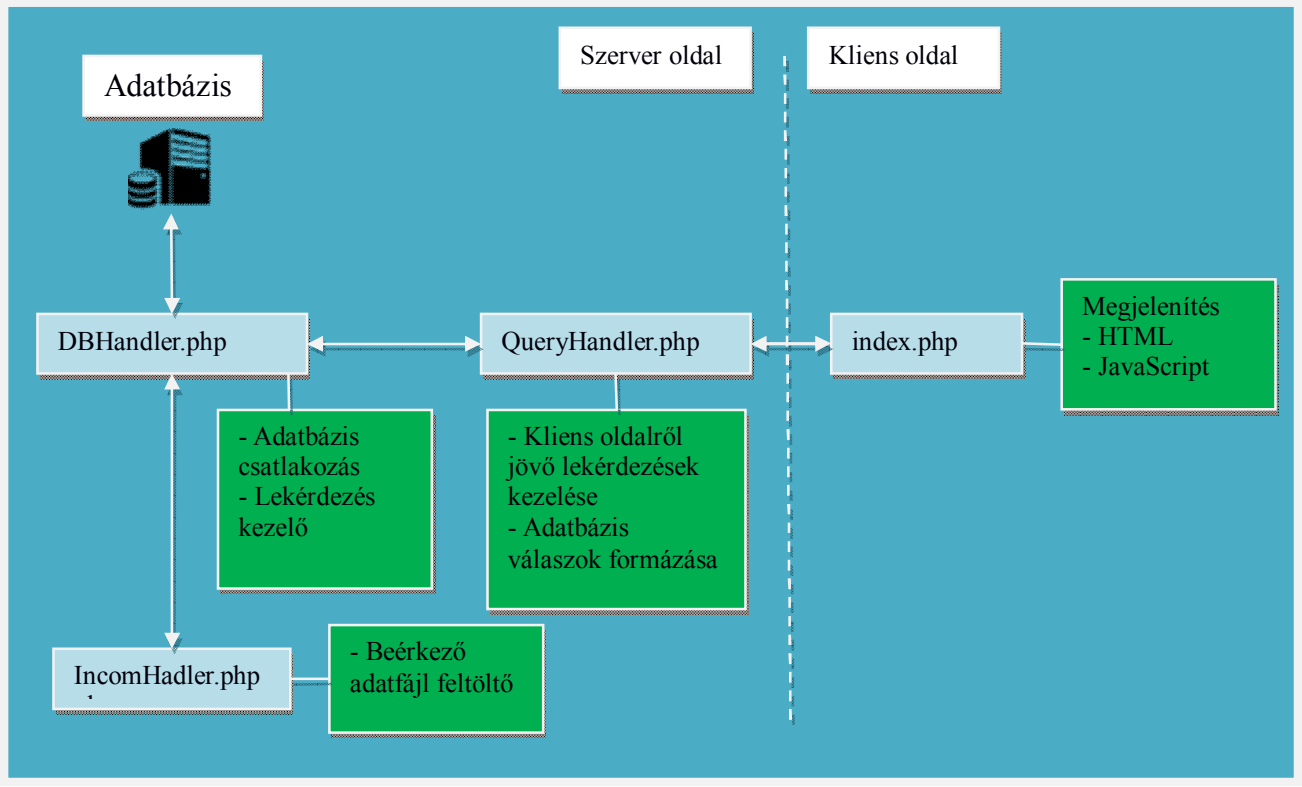
Az itt szereplő oszlopok neve (típusa) – tárolt jellemző magyarul:

- x_axis (float) – x koordináta
- y_axis (float) – y koordináta
- z_axis (float) – z koordináta
- id (int) – egyedi kulcs

A tábla minden oszlopára not null attribútum van beállítva.

5.6. PHP szkriptek áttekintése

Az általam megírt szerver és kliens oldali szkriptek kapcsolata látható 28. ábra. A következő fejezetekben a még nem ismertetett szkriptek kerülnek bemutatásra.



28. ábra – A szerver és kiens oldali programkapcsolatok

Az **index.php** fájlt leírása megtalálható az 5.4. fejezetben, itt nem kerül újra tárgyalásra.

5.6.1. Adatbázis kezelő – DBHandler.php

```php
//Kapcsolódás az adatbázishoz
$mysqli_connection=@mysqli_connect($DATABASE_HOST,$DATABASE_USER,$DATABASE_PW,$mysql_DB);

if (mysqli_connect_errno()) {
    //Nem sikerült a MySQL kiszolgálóval felvennei a kapcsolatot;
    f_error_log("Connecting to MySQL database is failed" , $ServerFolderError.$ServerErrorLogFileName);
}
else{
    mysqli_set_charset($mysqli_connection,'utf8');
    mysqli_query($mysqli_connection,"SET collation_connection = utf8_hungarian_ci");
}
```

29. ábra - Kapcsolódás MySQL adatbázishoz

A DBHandler szkriptben állíthatjuk be, hogy pontosan milyen adatbázishoz szeretnénk kapcsolódni. Itt adhatjuk meg az adatbázis elérési útját, nevét, felhasználói nevet és jelszót. A szkript meghívásakor kapcsolódik a megadott adatbázishoz. Ennek megvalósítása látható a 29. ábra. Továbbá itt található egy függvény mellyel le tudunk kérdezni az adatbázisból. Bemenő paramétere nincs.

5.6.2.Beérkező fájlok kezelése – IncomHandler.php

Az IncomHandler szkriptben az RPi-ktől beérkezett THPE és fluxus mérési eredményeket töltöm fel az adatbázisba. Ezt a szkriptet ütemezett feladatként kell futtatni majd a webszerveren, mely megadott időközönként megvizsgálja az INCOMING mappába feltöltött fájlokat. Az automatikus feldolgozás érdekében fájl elnevezési és fájl szerkezeti konvenciót vezettem be. A szkriptnek nincsen bemenő paramétere.

Az adatokat tartalmazó fájloknak az elnevezése kötött formátumú. A szkript a fájlok nevéből tudja, hogy melyik RPi küldte az adatot és hogy milyen típusú adatot küldött. Fluxus adattípusú fájloknál a küldési időt is a fájlnévbe kódoltam.

THPE esetében a fájl elnevezésében az RPi nevéhez kell hozzáfűzni a „_THPE” karaktereket. Egy példa erre, ha az adott RPi neve pl. „rpi_12”, akkor a fájl neve rpi_12_THPE lesz. Fluxus adatoknál a fájl nevében szerepelnie kell az időbélyeg információnak is. Formátuma a következő: rpinév___YYYY_M_DD__HHxMMxSS_FULX. Mindez szövegesen a következőképpen írható le: zz RPi neve után három alulvonás karakter következik, majd ezután jön a dátum, melynek egyes részei egy-egy alulvonás karakterrel vannak elválasztva, ezt követően két

alulvonás jön, majd az óra, perc, másodperc értékek szerepelnek, köztük „x” karakterrel. A végre pedig a „_FLUX” adattípus lejölés kell hozzáfűzni. Egy példa ennek alkalmazására RPi név: rpi_2, dátum: 2015.12.09 12:30:00, adattípusa: fluxus adat, esetén ekkor a fájlnév a következőképpen fog kinézni: rpi_2___2015_12_09__12x30x00_FLUX

A szkript az eredményfájlokat text formátumú fájlokban várja az RPi-ktől. A fájlokban minden sor egy adattábla bejegyzés lesz, ezért nem lehet benne üres sor. A sorokban szereplő értékeket tabulátorral kell elválasztani, a not null attribútumú oszlopoknál nem lehet üresen az adott oszlopban egyetlen érték sem.

A THPE típusú fájlokban egy sornak a következő mérési értékeket kell tartalmaznia ebben a sorrendben: hatásfok, külső páratartalom, belső páratartalom, külső hőmérséklet, belső hőmérséklet, külső nyomás, belső nyomás, időbélyeg. A fluxust tartalmazó fájlokban egy sor az alábbi értékeket kell, hogy tartalmazza: x koordináta, y koordináta, z koordináta.

```
if ($response == null){
    f_error_log("Cannot create thpe dara table for rpi  " .$sql,$ServerFolderError.$ServerErrorLogFileName);
}
else{
    #file beolvasás
    $sql = "LOAD DATA INFILE '".$ServerFolder.$THPEfile."' INTO TABLE ".$TempRPIname."_thpe_data;";
    $response = f_mySql_query($sql);
    usleep(500);

    if($response == null){
        f_error_log("Could not load data from file  " .$sql);
    }
    else{
        #Minden ok, volt, file átmozgatása a DONE mappába
        f_move_file($ServerFolder.$THPEfile, $ServerFolderDone.$THPEfile);
    }
}
```

30. ábra - Adatfeltöltés MySQL táblába külső fájlból

Az INCOMING mappában a szkript megvizsgál minden fájlt, amit az RPi-k feltöltöttek. Egy adott fájl esetében a fájl nevének feldolgozásából tudja a szkript, hogy THPE adatokat tartalmaz az adott fájl. Első körben a szkript lekérdezi, hogy az adatbázis thpe_lookup táblájában szerepel-e az Rpi neve, azaz, hogy küldött-e már valaha ilyen típusú adatot az adott RPi. Ekkor két eset lehetséges, az első, ha még nem küldött soha ilyen jellegű adatot, ekkor nem szerepel a thpe_lookup táblában.

```
#még nem szerepel az rpi listában
if (mysqli_fetch_array($response) ==  false) {
    #listához való hozzáadás
    $sql = "INSERT INTO thpe_lookup VALUES ('".$TempRPIname."');";

    $response = f_mySql_query($sql);
    usleep(500);

    if ($response == null){
        f_error_log("RPi name was not in the Rpi_lookup table, but could not insert it - sql: " .$sql , $ServerFolderError.$ServerErrorLogFileName);
    }
    else{
        #tábla kreálás CREATE TABLE
        $sql = "CREATE TABLE ".$TempRPIname."_thpe_data (efficiency     FLOAT NOT NULL,
                        Amb_H     FLOAT NOT NULL,
                        Inn_H     FLOAT NOT NULL,
                        Amb_T     FLOAT NOT NULL,
                        Inn_T     FLOAT NOT NULL,
                        Amb_P     FLOAT NOT NULL,
                        Inn_P     FLOAT NOT NULL,
                        Sample_Date datetime NOT NULL PRIMARY KEY);";
        $response = f_mySql_query($sql);
        usleep(500);
```

31. ábra - Tábla készítési MySQL parancs létrehozása

Ebben az esetben bejegyzi az RPi nevét a thpe_lookup táblába, majd létrehoz számára egy adat táblát az adatbázisban használva a névkonvenciós előírásokat. Erre láthatunk példát a 31. ábra. Majd ezután parancsot küld a MySQL-nek, hogy töltse be az adatokat a fájlból az adattáblába. Egy ilyen művelet látható a 30. ábra. A második esetben, azaz ha már szerepelt az RPi neve a thpe_lookup táblában, akkor nem ez az első eset, hogy adatot kapunk ettől az RPI-től, tehát létezik hozzá már adattábla. Az adattábla névkonvenciót alkalmazva összerakja a szkript az adattábla nevét majd parancsot küld a MySQL-nek, hogy töltse be az adatokat a fájlból az adattáblába. A szkript a feltöltés után a DONE mappába helyezi az adott fájlt. Ha bármely lekérdezésnél vagy parancsnál hiba történik, arról hibabejegyzést készít a szkript.

Fluxus adatok feltöltésének a menete hasonló a THPE adatfájl feldolgozásához. A szkript az INCOMING mappában a fájl nevéből tudja meg, hogy fluxus adatokat tartalmaz a adott fájl. Ezután a fájlnévből kinyeri az RPi nevét és a küldési dátumot. Ezen adatok birtokában lekérdezi a flux_lookup adattáblából, hogy az adott RPi bejegyzett-e már ilyen időbélyeggel adatot. Ha már létezik ilyen bejegyzés, akkor hibabejegyzést készít és áthelyezi a fájlt az ERROR mappába. Ha nem szerepelt a flux_lookup táblában az adott RPi-név és küldési dátum kombináció, akkor bejegyzi ezeket az értékeket a fux_lookup táblába, majd létrehoz egy adattáblát a névkonvenciós előírásokat alkalmazva az adatbázisban. Ezt követően parancsot küld a MySQL-nek, hogy töltse be az adatokat a fájlból az adattáblába. A szkript a feltöltés után a DONE mappába helyezi az adott fájlt. Ha bármely lekérdezésnél vagy parancsnál hiba történik, arról hibabejegyzést készít a szkript.

5.6.3. Lekérdezéskezelő – QueryHandler

```
elseif($TYPE == 3){
    //Akkor ez egy FLUX dátum kérés egy rpi -re
    $sql = "SELECT time_section FROM flux_lookup WHERE RPi_name = \"".$RPI."\" ORDER BY time_section";

    $response = f_mySql_query($sql);

    while($row = mysqli_fetch_array($response)){
            echo "</br>" . $row['time_section'] ;
            echo "<option value=\"". $row['time_section'] ."\">" . $row['time_section'] . "</option>";
    }
}
```

32. ábra - A QueryHandler a TYPE paraméter alapján építi fel a lekérdezést és fromázza a választ

A QueryHandler szkript felel, a weblapokról érkező kérések kezelésért, öt bemenő paraméterrel rendelkezik, amit híváskor meg kell, neki adni, ezek nevei és funkciói a következők:

- TYPE – Ebből a paraméterből tudja a szkript, hogy melyik weblap elem hívta meg. Ez alapján tudja, hogy mondjuk THPE kirajzolás kérés, vagy Fluxus rajzolás kérés érkezett be,
- RPI – a kiválasztott RPi neve,
- DATE_FROM – a lekérdezési időintervallum kezdete,
- DATE_TO – lekérdezési időintervallum vége,
- SENSOR – kiválasztott THPE szenzor betűjele,

A szkriptet JavaScript függvények hívják meg, melyek kliens oldalon kitöltik a paramétereit, részben előre beírt értékekkel (pl. TYPE paraméter), részben a HTML elemek aktuális értékeivel (pl. DATE_FROM paraméter). Erre láthatunk példát a 32. ábra A szkript összeállítja a lekérdezést, melyet a DBHandler adatbázis lekérdező függvényének használatával elküld a MySQL felé. Az adatbázisból érkező válasz feldolgozza és JSon formátumban adja vissza a kliens oldali JavaScriptnek.

5.6.4. Megjelenítés betöltő – PlotChart

```
<?php
####################################
# Chart megjelenítést vezérlő szkript#
####################################

$TYPE =$_REQUEST['TYPE'];

    if($TYPE == 1){
        echo "<div class=\"center\" id=\"container\" style=\"min-width: 310px;height:400px;border:1px solid blueq\"></div></br>";
    }
    elseif($TYPE == 2){
        echo "<div class=\"center\" id=\"container_flux\" style=\"min-width: 1000px;height:1000px;border:1px solid blueq\"></div></br>";
    }

?>
```

33. ábra - A PlotChart szkript TPYE paraméter szerint más HTML elembe tölti az adott

Ez egy igen rövid szkript, aminek egy bemenő paramétere van, a TYPE. A szkript a kirajzolni kívánt diagramot tölti bele a TYPE függvényében vagy a THPE_MEGJENEITES vagy a FLUX_MEGJENEITES ID-val rendelkező div típusú HTML objektumba. A teljes kód a 33. ábra látható.

5.7. Teszteredmények

A tesztfájlokkal teszteltem a feltöltés és a megjelenítéshez írt lekérdezéseket. A rendszer jól működött, a tesztelés eredményei a már korábban bemutatott 34., 35. és 23. ábrákon láthatóak.

Összefoglaló

Diplomadolgozatom készítése során rengeteg új ismeretet szereztem. Többek között a detektorfizika és azon belül is a müontomográfia területéről. A REGARD csoporttal együttműködve sikeresen terveztem és implementáltam egy olyan rendszert, mely müontomográfok segítségével vulkánok online monitorozására ad lehetőséget.

Megterveztem és legyárttattam egy hőmérséklet, páratartalom, nyomás mérésére alkalmas kártyát és kidolgoztam a program felépítését. A tervezés során figyelembe vettem a kutatócsoport igényeit, köztük az egyik legfontosabbat, hogy a kártya illeszthető legyen a tomográf busz rendszerébe.

Megismerkedtem egy GSM/GPRS modul használatával és ezt a müontomográf mellett található Raspberry Pi PC-hez illesztettem. A GSM/GPRS modul alkalmazásával mobil internetkapcsolatot létesítettem, és így már egy általam Python programnyelven írt program segítségével az autonóm tomográf képes volt feltölteni adatait egy távoli szerverre.

A szerver oldalon PHP szkript segítségével MySQL adatbázisba mentettem a mérési eredményeket, fejlesztettem egy weboldalt is, ami PHP, HTML, JavaScript programnyelvek együttes használatával képes a mérési eredményeket megjeleníteni. Szakmailag nagyon hasznos volt számomra megismerkedni ezekkel a nyelvekkel, és betekintést nyerni a szerver oldali programozás világába.

A rendszer a teszteken jól szerepelt, és a jövőben várható a tesztelés kiterjesztése az éles rendszerre is.

Köszönetnyilvánítás

Ezúton szeretnék köszönetet mondani mindazoknak, akik segítettek eme diplomamunka elkészítése során. Köszönöm szüleimnek, páromnak, barátaimnak, akik nélkül ma nem tartanék itt ahol. Köszönöm páromnak, aki mindig támogatott, és segített a magyar nyelv útvesztőiben eligazodni. Köszönöm a Prolan cég dolgozóinak, akik szakmai és emberi támogatásukról biztosítottak. Köszönet illeti a REGARD csoport tagjait, különös képen köszönöm Dr. Hamar Gergő, Dr. Oláh László és Dr. Varga Dezsőnek, hogy mindig gyorsan és szívesen segítettek, ha bármilyen problémába ütköztem.
Külön kiemelném belső konzulensemet Dr. Hídvégi Timót tanárurat, akitől szakmailag nagyon sokat tanultam, és végig támogatott dolgozatom megírása során. Külön kiemelném még külső konzulensemet, Dr. Hamar Gergőt, akitől rengeteg segítséget kaptam, tőle ismerhettem meg a detektor fizika alapjait, és nagyon sokat segített a rendszer megismerésében. Köszönöm mind belső és külső konzulensem áldozatos munkáját, észrevételeikkel jelentősen hozzájárultak a dolgozatom minőségének javításához!

Irodalomjegyzék

[1] http://regard.kfki.hu/?page=projects&sub=MT 2015. szeptember 23.

[2] Oláh László: Szerkezetvizsgálat kozmikus részecskék segítségével,
MSc szakdolgozat ELTE TTK, Fizikus szak, 2012

[3] Hőmérsékletmérő szenzor adatlap:
http://ww1.microchip.com/downloads/en/DeviceDoc/25095A.pdf 2015.10.22

[4] Páratartalom mérő szenzor adatlap: http://sensing.honeywell.com/honeywell-sensing-humidicon-hih6100-series-product-sheet-009059-6-en.pdf?name=HIH6130-021-001

2015.10.22

[5] Nyomásmérő szenzor adatlap: https://ae-bst.resource.bosch.com/media/products/dokumente/bmp180/BST-BMP180-DS000-12~1.pdf

2015.10.22

[6] Mikrokontroller adatlap: http://ww1.microchip.com/downloads/en/DeviceDoc/60001168H.pdf

2015.10.22

[7] Feszültség szabályozó adatlap: http://ww1.microchip.com/downloads/en/DeviceDoc/22026b.pdf

2015.10.22

[8] https://www.raspberrypi.org/products/raspberry-pi-2-model-b/ 2015.11.30.

[9] https://www.raspberrypi.org/downloads/ 2015.11.30.

[10] https://learn.adafruit.com/downloads/pdf/adafruit-fona-mini-gsm-gprs-cellular-phone-module.pdf (Last updated on 2015-11-18 09:00:17AM EST) 2015.11.30.

[11] http://www.adafruit.com/datasheets/sim800h_hardware_design_v1.00.pdf 2015.11.30

[12] https://electrosome.com/uart-raspberry-pi-python/ 2015.11.30

[13] http://linux.die.net/man/8/chat 2015.12.01

[14] https://learn.adafruit.com/fona-tethering-to-raspberry-pi-or-beaglebone-black/overview
2015.12.01

[15] https://raw.githubusercontent.com/adafruit/FONA_PPP/master/fona 215.12.01

[16] https://en.wikipedia.org/wiki/XAMPP 2015.12.04

[17] https://www.apachefriends.org/index.html 2015.12.04

[18] https://hu.wikipedia.org/wiki/HTML 2015.12.04

[19] https://en.wikipedia.org/wiki/HTML 2015.12.04

[20] https://en.wikipedia.org/wiki/World_Wide_Web 2015.12.04

[21] https://en.wikipedia.org/wiki/MySQL 2015.12.04

[22] https://hu.wikipedia.org/wiki/PHP 2015.12.04

[23] https://en.wikipedia.org/wiki/JavaScript 2015.12.04

[24] https://en.wikipedia.org/wiki/JQuery 2015.12.04

[25] https://hu.wikipedia.org/wiki/Ajax_%28programoz%C3%A1s%29 2015.12.04

[26] http://www.highcharts.com/ 2015.12.07

[27] K. Nagamine et al: Method of probing inner-structure of geophysical substance with the horizontal cosmic-ray muons and possible application to volcanic eruption prediction

NIM A 356, 585-595 1995

[28] H.Tanaka et al: Development of a portable assembly-type cosmic-ray muon module for measuring the density structure of a column of magma

Earth Planets Space ,62, 119–129, 2010

[29] F. Sauli:

Principles of operation of multiwire proportional and drift chambers

Lectures given in the Academic Training Programme of CERN 1975-1976 (1977)

[30] D.Varga, G.Hamar, G. Kiss:

Asymmetric Multi-Wire Proportional Chamber with reduced requirements to mechanical precision

NIM A 648, p 163-167 2011

[31] D. Varga, G. Kiss, G. Hamar, Gy. Bencédi:

Close cathode chamber: Low material budget MWPC

NIM A 698, p 11-18 2013.01.

Mellékletek

5.8. A kifejlesztett PHP, JavaScript, HTML programkódok

5.8.1. Index.php

```
<html>
	<!-- Kliens oldali weboldal -->
    <head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
    <title>Thesis</title>
    </head>

	<style>
	.center {
		margin: auto;
		width: 60%;
		padding: 10px;
	</style>

	<body>

	 <script src="jquery.js"></script>
	 <script src="json2.js"></script>
    <script src="//ajax.googleapis.com/ajax/libs/jquery/1.8.2/jquery.min.js"></script>
    <script>window.jQuery || document.write('<script src="js/jquery-1.8.3.min.js"><\/script>')</script>
    <script src="http://code.highcharts.com/highcharts.js"></script>
	 <script src="https://code.highcharts.com/modules/heatmap.js"></script>
    <script src="http://code.highcharts.com/highcharts-more.js"></script>
    <script src="http://code.highcharts.com/modules/exporting.js"></script>

<script>

	//Chart rajzolás
		function drawTHPEChart (p_time_x_arr,
                                                        p_x_title_str,
                                                        p_y_title_str,
                                                        p_tooltipsuffix_title_str,
                                                        p_y_data_1_arr,
                                                        p_y_data_2_arr,
```

```
                                                    p_y_data_1_title_str,
                                                    p_y_data_2_title_str,
                                                    p_main_title_str,
                                                    p_sub_title_str) {
$('#container').highcharts({
    title: {
        text: p_main_title_str,
        x: -20 //center
    },
    subtitle: {
        text: p_sub_title_str,
        x: -20
    },
    xAxis: {
                                        title: {
            text: p_x_title_str
        },
        categories: p_time_x_arr
    },
    yAxis: {
        title: {
            text: p_y_title_str
        },
        plotLines: [{
            value: 0,
            width: 1,
            color: '#808080'
        }]
    },
    tooltip: {
        valueSuffix: p_tooltipsuffix_title_str
    },
    legend: {
        layout: 'vertical',
        align: 'right',
        verticalAlign: 'middle',
        borderWidth: 0
    },
                                plotOptions: {
```

```
                                    line: {
                                        dataLabels: {
                                            enabled: false
                                        },
                                        enableMouseTracking: true
                                    }
                                },
        series: [{
            name: p_y_data_1_title_str,
            data: p_y_data_1_arr
        }, {
            name: p_y_data_2_title_str,
            data: p_y_data_2_arr
        }]
    });
};

    //Flux heatmap kirajzolása
    function drawFluxHeatMap (p_x_cat_arr, p_y_cat_arr, p_xyz_data_arr,p_text) {

            $('#container_flux').highcharts({

                chart: {
                    type: 'heatmap',
                    marginTop: 40,
                    marginBottom: 80,
                    plotBorderWidth: 1
                },
                title: {
                    text: p_text
                },

                xAxis: {
                    categories: p_x_cat_arr
                },

                yAxis: {
                    categories: p_y_cat_arr,
                    title: null
```

```
        },

        colorAxis: {
            min: 0,
                //   minColor: '#0000FF' ,
                //   maxColor: '#FF0000'
                minColor: '#00A8FF',
                maxColor: '#FF0000'
            },

            legend: {
                align: 'right',
                layout: 'vertical',
                margin: 0,
                verticalAlign: 'top',
                y: 25,
                symbolHeight: 280
            },

            tooltip: {
                formatter: function () {
                    return '<b>'+ '</b>  x: <b>' + this.series.xAxis.categories[this.point.x]  +
                        '<br></b> y: <b>' + this.series.yAxis.categories[this.point.y] +  '<br></b> z: <b>' +
this.point.value;
                }
            },

            series: [{
                name: 'Flux distribution',
                borderWidth: 1,
                data: p_xyz_data_arr,
                dataLabels: {
                    enabled: false,
                    color: '#000000'
                }
            }]

        });
    }
```

```
function testUploadBtn(){
                var JQueryAjax = $.post('IncomHandler.php')
                .done(function(data) {
                        $('#TEST_MEGJELENITES').html(data)
                })
}

//THPE dátum feltöltés
function fillRpiTHPEDates(){

        var JS_selectedRPI = $('#selectrpi_THPE').val()
                //átpasszuljuk a kérést php nak, az eredményt populáljuk majd a select- html résznek.
                //var JQueryAjax = $.post(
'QueryHandler.php?DATE='+JS_measuremenst+'&SENSOR='+JS_selectTHP+'&FIRST='+JS_first_call)
                var JQueryAjax = $.post(
'QueryHandler.php?TYPE='+1+'&RPI='+JS_selectedRPI+'&DATE_FROM='+0+'&DATE_TO='+0+'&SENSOR='+0)
                .done(function(data) {
                        $('#select_THPE_FROM').html(data)
                        $('#select_THPE_TO').html(data)
                })

}

//FLUX dátum feltöltés
function fillRpiFluxDates() {
        var JS_selectedRPI = $('#selectrpi_FLUX').val()
        if (JS_selectedRPI != "A_selected") {
                //RPi-t választott
                //átadjuk a kérést php nak, az eredményt populáljuk majd a select- html résznek.
                var JQueryAjax = $.post(
'QueryHandler.php?TYPE='+3+'&RPI='+JS_selectedRPI+'&DATE_FROM='+0+'&DATE_TO='+0+'&SENSOR='+0)
                .done(function(data) {
                        $('#select_FLUX_FROM').html(data)
                        $('#select_FLUX_TO').html(data)
                })
        }
        else{
                //Aggregáltat választott
                //átpasszuljuk a kérést php nak, az eredményt populáljuk majd a select- html résznek.
```

```
                var JQueryAjax = $.post(
'QueryHandler.php?TYPE='+4+'&RPI='+JS_selectedRPI+'&DATE_FROM='+0+'&DATE_TO='+0+'&SENSOR='+0)
                .done(function(data) {
                        $('#select_FLUX_FROM').html(data)
                        $('#select_FLUX_TO').html(data)
                })
        }
}

//FLUX kirajzolás
function showFluxHeatMap(){

        var JS_Time_from = $('#select_FLUX_FROM').val()
        var JS_Time_to = $('#select_FLUX_TO').val()
        var JS_SelectedRPI = $('#selectrpi_FLUX').val()

        //null input vizsgálat
        if (JS_SelectedRPI == null) {
                alert("RPi must be selected!")
                return
        }
        else if (JS_Time_from == null) {
                alert("Start date must be selected!")
                return
        }
        else if (JS_Time_to == null) {
                alert("End date must be selected!")
                return
        }

        //DATE vizsgálat
        var JS_Time_from_CONV=JS_Time_from.replace(/ /g,"T")
        date_from = new Date(JS_Time_from_CONV)

        var JS_Time_to_CONV=JS_Time_to.replace(/ /g,"T")
        date_to = new Date(JS_Time_to_CONV)


        if (date_from > date_to) {
```

```
                alert("Start date must be less or equal than end date!")
                return
        }

        //Rajzoláshoz kellő konténerek
        var var_x_arr = []
        var var_y_arr = []
        var var_z_arr = []

        var var_tmp_array = []

        if (JS_SelectedRPI != 'A_selected') {
                //Akkor rpi flux ot választott
                var JQueryAjax = $.post(
'QueryHandler.php?TYPE='+5+'&RPI='+JS_SelectedRPI+'&DATE_FROM='+JS_Time_from+'&DATE_TO='+JS_Tim
e_to+'&SENSOR='+0)
                .done(function(data) {
                        data = JSON.parse(data)

                for (var i = 0; i < data.x.length; i++) {
                        var_x_arr[i] = parseFloat(data.x[i])
                }

                for (var i = 0; i < data.y.length; i++) {
                        var_y_arr[i] = parseFloat(data.y[i])
                }

                for (var i = 0; i < data.z.length; i++) {
                        var_z_arr[i] = parseFloat(data.z[i])
                }

                var k = 0
                for (var i =0; i < (data.x.length/3); i++) {
                        for (var j = 0; j < (data.y.length/3); j++) {
                                var_tmp_array[k] = [i,j,parseFloat(var_z_arr[k])]
                                k = k+1
                        }
                }

                //text összerakás
```

```
                JS_SelectedRPI = JS_SelectedRPI.replace("r","R")
                JS_SelectedRPI = JS_SelectedRPI.replace("p","P")
                JS_SelectedRPI = JS_SelectedRPI.replace("_"," ")

                var var_text = 'Flux map: '+JS_SelectedRPI+' - '+JS_Time_from +' - '+ JS_Time_to

                var JQueryAjax = $.post('PlotChart.php?TYPE='+2).done(function(data){
                        $('#FLUX_MEGJELENITES').html(data);
                                drawFluxHeatMap (var_x_arr,
                                                                var_y_arr,
                                                                var_tmp_array,
                                                                var_text)
                                })
                })
        }
        else{
                //Aggregáltat választott
                var JQueryAjax = $.post(
'QueryHandler.php?TYPE='+6+'&RPI='+JS_SelectedRPI+'&DATE_FROM='+JS_Time_from+'&DATE_TO='+JS_Tim
e_to+'&SENSOR='+0)
                .done(function(data) {
                                data = JSON.parse(data)

                for (var i = 0; i < data.x.length; i++) {
                        var_x_arr[i] = parseFloat(data.x[i])
                }

                for (var i = 0; i < data.y.length; i++) {
                        var_y_arr[i] = parseFloat(data.y[i])
                }

                for (var i = 0; i < data.z.length; i++) {
                        var_z_arr[i] = parseFloat(data.z[i])
                }

                var k = 0
                for (var i = 0; i < (data.x.length/3); i++) {
                        for (var j = 0; j < (data.y.length/3); j++) {
                                var_tmp_array[k] = [i,j,parseFloat(var_z_arr[k])]
                                k = k+1
```

```
                    }
                }

                var var_text = 'Flux map: aggregated  -  '+JS_Time_from +' - '+ JS_Time_to

                var JQueryAjax = $.post('PlotChart.php?TYPE='+2).done(function(data){
                    $('#FLUX_MEGJELENITES').html(data);
                        drawFluxHeatMap (var_x_arr,
                                                var_y_arr,
                                                var_tmp_array)

                        })
                })
        }

    }

    //THPE kirajzolás
    function showTHPEChart(){

/* Változó beolvasás: különböző id-val rendelkező text-mezőket egy-egy változóba*/
        var JS_selectedRPI = $('#selectrpi_THPE').val()
        var JS_selectTHPE = $('#selectTHPE').val()
        var JS_Time_from = $('#select_THPE_FROM').val()
        var JS_Time_to = $('#select_THPE_TO').val()


        //null input vizsgálat
        if (JS_selectedRPI == null) {
            alert("RPi must be selected!")
            return
        }
        else if (JS_selectTHPE == null) {
            alert("Sensor must be selected!")
            return
        }
        else if (JS_Time_from == null) {
            alert("Start date must be selected!")
            return
```

```
}
else if (JS_Time_to == null) {
	alert("End date must be selected!")
	return
}

//DATE vizsgálat
var JS_Time_from_CONV=JS_Time_from.replace(/ /g,"T")
date_from = new Date(JS_Time_from_CONV)

var JS_Time_to_CONV=JS_Time_to.replace(/ /g,"T")
date_to = new Date(JS_Time_to_CONV)


if (date_from >= date_to) {
	alert("Start date must be less than end date!")
	return
}


//Konvertálások - ahol kell
//THP sensor
if (JS_selectTHPE == "T_selected") {
	JS_selectTHPE = 1
}
else if (JS_selectTHPE == "H_selected" ) {
	JS_selectTHPE = 2
}
else if (JS_selectTHPE == "P_selected") {
	JS_selectTHPE = 3
}
else{
	JS_selectTHPE = 4
}

/* AJAX hívás -> átküldi a szerverre és az visszaadja a feldolgozott eredményt/eredményeket */
var JQueryAjax = $.post(
'QueryHandler.php?TYPE='+2+'&RPI='+JS_selectedRPI+'&DATE_FROM='+JS_Time_from+'&DATE_TO='+JS_Tim
e_to+'&SENSOR='+JS_selectTHPE)
.done(function(data) {
```

```
var var_time_x_arr = []
var var_x_title_str = 'Elapsed time from start date'
var var_y_title_str = ''
var var_tooltipsuffix_title_str = ''
var var_y_data_1_arr = []
var var_y_data_2_arr = []
var var_y_data_1_title_str = ''
var var_y_data_2_title_str = ''
var var_main_title_str = ''
var var_sub_title_str = ''

console.log("hello")
console.log(data)
data = JSON.parse(data)
console.log(data)
console.log(data[0].Sample_Date)

if (JS_selectTHPE == 1){

        for (var i = 0; i < data.length; i++) {
                var_time_x_arr[i] = String(data[i].Sample_Date)
                var_y_data_1_arr[i] = parseFloat(data[i].Amb_T)
                var_y_data_2_arr[i] = parseFloat(data[i].Inn_T)
        }

        var_y_title_str = 'Temperature (°C)'
        var_tooltipsuffix_title_str = '°C'
        var_y_data_1_title_str = 'Ambient temperature'
   var_y_data_2_title_str = 'Inner temperature'
        var_main_title_str = 'Selected time scope'
        var_sub_title_str = JS_Time_from + " - " + JS_Time_to
}
else if (JS_selectTHPE == 2) {
        for (var i = 0; i < data.length; i++) {
                var_time_x_arr[i] = String(data[i].Sample_Date);
                var_y_data_1_arr[i] = parseFloat(data[i].Amb_H)
                var_y_data_2_arr[i] = parseFloat(data[i].Inn_H)
        }
```

```
            var_y_title_str = 'Humidity (%)'
            var_tooltipsuffix_title_str = ' %'
            var_y_data_1_title_str = 'Ambient humidity'
    var_y_data_2_title_str = 'Inner humidity'
            var_main_title_str = 'Selected time scope'
            var_sub_title_str = JS_Time_from + " - " + JS_Time_to
}
else if (JS_selectTHPE == 3){
            for (var i = 0; i < data.length; i++) {
                        var_time_x_arr[i] = String(data[i].Sample_Date);
                        var_y_data_1_arr[i] = parseFloat(data[i].Amb_P)
                        var_y_data_2_arr[i] = parseFloat(data[i].Inn_P)
            }

            var_y_title_str = 'Pressure (mBar)'
            var_tooltipsuffix_title_str = ' mBar'
            var_y_data_1_title_str = 'Ambient pressure'
    var_y_data_2_title_str = 'Inner pressure'
            var_main_title_str = 'Selected time scope'
            var_sub_title_str = JS_Time_from + "  -  " + JS_Time_to
}
else{

            for (var i = 0; i < data.length; i++) {
                        var_time_x_arr[i] = String(data[i].Sample_Date);
                        var_y_data_1_arr[i] = parseFloat(data[i].efficiency)
            }
            JS_selectedRPI = JS_selectedRPI.replace("r","R")
            JS_selectedRPI = JS_selectedRPI.replace("p","P")
            JS_selectedRPI = JS_selectedRPI.replace("_"," ")

            var_y_title_str = 'Efficiency'
            var_tooltipsuffix_title_str = ' '
            var_y_data_1_title_str = JS_selectedRPI + 'efficiency'
            var_y_data_2_title_str = 'none'
            var_main_title_str = 'Selected time scope'
            var_sub_title_str = JS_Time_from + "  -  " + JS_Time_to
```

```
                }

                var JQueryAjax = $.post('PlotChart.php?TYPE='+1).done(function(data){
                        $('#THPE_MEGJELENITES').html(data);
                                drawTHPEChart (         var_time_x_arr,
                                                        var_x_title_str,
                                                        var_y_title_str,
                                                        var_tooltipsuffix_title_str,
                                                        var_y_data_1_arr,
                                                        var_y_data_2_arr,
                                                        var_y_data_1_title_str,
                                                        var_y_data_2_title_str,
                                                        var_main_title_str,
                                                        var_sub_title_str)
                })

        })
    }

</script>

<!-- THPE rész -->
                <br><b>Scheduled task test</b></br>
                <input id="TEST_UPLOAD_BTN" type="button" value="Test upload" onclick="testUploadBtn()"
style="border:0px #000 solid; background-color:#ACD65E; color:#FFF; padding:5px 10px 5px 10px; border-
radius:5px; cursor:pointer; " /></br></br>
                <br><b>Select data to show T,H,P,E graph:</b></br>
                <select id="selectrpi_THPE" onchange="fillRpiTHPEDates()">
                        <option value="" disabled selected>Select RPi</option>
                <?php
                        include_once('..\WebServ\DBHandrel.php');

                        $sql = "SELECT RPi_name FROM thpe_lookup ORDER BY RPi_name";

                        $response = f_mySql_query($sql);

                        while($row = mysqli_fetch_array($response)){
                                echo "</br>" . $row['RPi_name'] ;
                                echo "<option value=\"". $row['RPi_name'] ."\">" . $row['RPi_name'] . "</option>";
                        }
```

```
		?>
		</select>
		<select id="selectTHPE">
			<option value="" disabled selected>Select data type</option>
			<option value="T_selected">Temperature</option>
			<option value="H_selected">Humidity</option>
			<option value="P_selected">Pressure</option>
			<option value="E_selected">Efficiency</option>
		</select> from
		<select id="select_THPE_FROM">
			<option value="" disabled selected>Select from date</option>
		</select> to
		<select id="select_THPE_TO">
			<option value="" disabled selected>Select to date</option>
		</select>
		<input id="SHOW_THPE_CALL_BTN" type="button" value="Show" onclick="showTHPEChart()"
style="border:0px #000 solid; background-color:#ACD65E; color:#FFF; padding:5px 10px 5px 10px; border-
radius:5px; cursor:pointer; " /></br></br>


<!-- FLUX rész -->
<br><b>Select data to show flux map:</b></br>
		<select id="selectrpi_FLUX" onchange="fillRpiFluxDates()">
			<option value="" disabled selected>Select RPi</option>
		<?php
			include_once('..\WebServ\DBHandrel.php');

			$sql = "SELECT RPi_name FROM flux_lookup group by RPi_name ORDER BY
RPi_name";

			$response = f_mySql_query($sql);

			while($row = mysqli_fetch_array($response)){
				echo "</br>" . $row['RPi_name'] ;
				echo "<option value=\"". $row['RPi_name'] ."\">" . $row['RPi_name'] . "</option>";
			}
		?>
			<option value="A_selected">Aggregated</option>
		</select>  from
		<select id="select_FLUX_FROM">
			<option value="" disabled selected>Select from date</option>
```

```
                </select>  to
                <select id="select_FLUX_TO">
                        <option value="" disabled selected>Select to date</option>
                </select>
                <input id="SHOW_FLUX_CALL_BTN" type="button" value="Show"
onclick="showFluxHeatMap()" style="border:0px #000 solid; background-color:#ACD65E; color:#FFF; padding:5px
10px 5px 10px; border-radius:5px; cursor:pointer; " /></br></br>
<!-- MEGJELENÍTÉS rész -->
                <div id="THPE_MEGJELENITES"><br></br>
                <div id="FLUX_MEGJELENITES"><br></br>
                <div id="TEST_MEGJELENITES">
   </body>
</html>
```

5.8.2. QueryHandler.php

```
<?php
##############################################
# Kliens oldalról érkezo lekérdezéseket kezelo szkript #
##############################################

include_once('..\WebServ\DBHandrel.php');

   # $_REQUEST-el olvasom be az átküldött értékeket
         $TYPE =$_REQUEST['TYPE'];
         $RPI =$_REQUEST['RPI'];
         $DATE_FROM = $_REQUEST['DATE_FROM'];
         $DATE_TO = $_REQUEST['DATE_TO'];
   $SENSOR = $_REQUEST['SENSOR'];

   $sql_Amb_Sensor = "";
   $sql_Inn_Sensor = "";

   if($SENSOR == 1){
      $sql_Amb_Sensor = "Amb_T,";
      $sql_Inn_Sensor = "Inn_T";
   }
   elseif ($SENSOR == 2) {
      $sql_Amb_Sensor = "Amb_H,";
      $sql_Inn_Sensor = "Inn_H";
   }
```

```
    elseif($SENSOR == 3){
        $sql_Amb_Sensor = "Amb_P,";
        $sql_Inn_Sensor = "Inn_P";
    }
    else{
                $sql_Inn_Sensor = "efficiency";
        }

        #a dátumot átalakítja: 2015-12-04 10:12:01 --> 2015_12_04_10_12_01
        function trim_date($str){

                //kiszedjük a "-" -t lecseréljük "_"-ra
                $str = str_replace("-","_",$str);
                //a szóközt "_"-ra cseréljük
                $str = str_replace(" ","_",$str);
                //kiszedjük a 10:00:00 - ból a :-ket
                $str = str_replace(":","_",$str);
                return $str;
        }

        if($TYPE == 1){
                //Akkor ez egy THPE dátum kérés
                $sql = "SELECT Sample_Date FROM ".$RPI."_thpe_data ORDER BY Sample_Date";

                $response = f_mySql_query($sql);

                while($row = mysqli_fetch_array($response)){
                                echo "</br>" . $row['Sample_Date'] ;
                                echo "<option value=\"". $row['Sample_Date'] ."\">" . $row['Sample_Date'] .
"</option>";
                }
        }
        elseif($TYPE == 2){
                //Ez egy  THPE rajazolási kérés lesz
                $sql = "SELECT Sample_Date,".$sql_Amb_Sensor.$sql_Inn_Sensor." FROM ".$RPI."_thpe_data
WHERE (Sample_Date >= '".$DATE_FROM."') && (Sample_Date <= '".$DATE_TO."') ORDER BY Sample_Date";
                $response = f_mySql_query($sql);

                $rows = array();
```

```
		while($r = mysqli_fetch_assoc($response)) {
			$rows[] = $r;
		}

		echo json_encode($rows);
	}
	elseif($TYPE == 3){
		//Akkor ez egy FLUX dátum kérés egy rpi -re
		$sql = "SELECT time_section FROM flux_lookup WHERE RPi_name = \"".$RPI."\" ORDER BY
time_section";

		$response = f_mySql_query($sql);

		while($row = mysqli_fetch_array($response)){
				echo "</br>" . $row['time_section'] ;
				echo "<option value=\"". $row['time_section'] ."\">" . $row['time_section'] .
"</option>";
		}
	}
	elseif($TYPE == 4){
		//Akkor ez egy aggregált flux dátum kérés
		$sql = "SELECT time_section FROM flux_lookup ORDER BY time_section";

		$response = f_mySql_query($sql);

		while($row = mysqli_fetch_array($response)){
				echo "</br>" . $row['time_section'] ;
				echo "<option value=\"". $row['time_section'] ."\">" . $row['time_section'] .
"</option>";
		}
	}
	elseif($TYPE == 5){
		//Akkor ez egy RPi flux kérés

		//Nehogy legyen bentragadt temp tábla eldobjuk a táblát ha létezik
		$sql = "DROP TABLE IF EXISTS _temp_flux";
		$response = f_mySql_query($sql);

		//temp tábla készítése
		$sql = "CREATE TABLE _temp_flux ( x_axis FLOAT NOT NULL, y_axis FLOAT NOT NULL,
```

```
z_axis FLOAT NOT NULL, id INT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY )";
		$response = f_mySql_query($sql);


		//Lekéjkük, hogy milyen dátumok esnek a kiválasztott intervallumba
		$sql = "SELECT time_section FROM flux_lookup WHERE (time_section >= '".$DATE_FROM."')
&& (time_section <= '".$DATE_TO."') ORDER BY time_section";
		$response = f_mySql_query($sql);


		// Beletesszük a temp táblába a kiválasztott intervallumban lévo táblaadatokat
		while($row = mysqli_fetch_array($response)){
				$sql = "INSERT IGNORE INTO _temp_flux SELECT * FROM
".$RPI."_".trim_date($row['time_section']);
				f_mySql_query($sql);
		}


		// TEMP át kell konvertálni az oszlopokakt FLOAT-ról DECIMAL-ra, mert kiegészíti szeméttel a
számokat a MySQL
		$sql = "ALTER TABLE _temp_flux MODIFY x_axis DECIMAL(10,5)";
		f_mySql_query($sql);
		$sql = "ALTER TABLE _temp_flux MODIFY y_axis DECIMAL(10,5)";
		f_mySql_query($sql);
		$sql = "ALTER TABLE _temp_flux MODIFY z_axis DECIMAL(10,5)";
		f_mySql_query($sql);


		$x_array = array();
		$y_array = array();
		$z_array = array();


		//lekérjük az X tengely koordinátáit
		$sql = "SELECT x_axis FROM _temp_flux GROUP BY x_axis";
		$response = f_mySql_query($sql);


		//lementjük az x_array ba
		$idx = 0;
		while($row = mysqli_fetch_array($response)){


				if ($row == null){
					echo "X_NULL";
				}
				$x_array[$idx] = $row['x_axis'];
```

```
                $idx = $idx+1;
        }

        //lekérjük az Y tengely koordinátáit
        $sql = "SELECT y_axis FROM _temp_flux GROUP BY y_axis";
        $response = f_mySql_query($sql);

        //lementjük az y_array ba
        $idx = 0;
        while($row = mysqli_fetch_array($response)){

                if ($row == null){
                        echo "Y_NULL";
                }

                $y_array[$idx] = $row['y_axis'];
                $idx = $idx+1;
        }

        $idx = 0;
        foreach($x_array as $x_value){
                foreach($y_array as $y_value){
                usleep(100);
                $sql = "SELECT sum(z_axis) FROM  _temp_flux WHERE (x_axis =".$x_value. ")
AND (y_axis =".$y_value.");";
                $response = f_mySql_query($sql);

                if ($response == null){
                        $row[0] = 0;
                }
                else{
                        $row = mysqli_fetch_array($response);
                }

                if($row[0] == null){
                        $z_array[$idx] = 0;
                }
                else{
                        $z_array[$idx] = $row[0];
```

```
                        }

                        $idx = $idx + 1;
                }
                //echo"</br> DONE";
        }
        //echo"</br> ALL DONE</br>";

        /*$myfile = fopen("newfile.txt", "w") or die("Unable to open file!");

        $idx = 0;
        foreach($x_array as $x_value){
                foreach($y_array as $y_value){

                        $txt = $x_value.",".$y_value.",".$z_array[$idx]."\n";
                        $idx = $idx +1;
                        fwrite($myfile, $txt);
                }
        }

        fclose($myfile);*/

        echo json_encode(array('x' => $x_array, 'y' => $y_array, 'z' => $z_array));

        //A végén eldobjuk a táblát
        $sql = "DROP TABLE IF EXISTS _temp_flux";
        f_mySql_query($sql);

}
elseif($TYPE == 6){
        //Akkor ez egy agregált RPi flux lekérdezés
        //Nehogy legyen bentragadt temp tábla eldobjuk a táblát ha létezik
        $sql = "DROP TABLE IF EXISTS _temp_flux";
        $response = f_mySql_query($sql);

        //temp tábla készítése
        $sql = "CREATE TABLE _temp_flux ( x_axis FLOAT NOT NULL, y_axis FLOAT NOT NULL,
z_axis FLOAT NOT NULL, id INT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY )";
        $response = f_mySql_query($sql);
```

```
		//Lekéjkük, hogy milyen dátumok esnek a kiválasztott intervallumba
		$sql = "SELECT RPi_name, time_section FROM flux_lookup WHERE (time_section >= '".$DATE_FROM."') && (time_section <= '".$DATE_TO."') ORDER BY time_section";
		$response = f_mySql_query($sql);

		// Beletesszük a temp táblába a kiválasztott intervallumban lévo táblaadatokat
		while($row = mysqli_fetch_array($response)){
			$sql = "INSERT IGNORE INTO _temp_flux SELECT * FROM ".$row['RPi_name']."_".trim_date($row['time_section']);
			f_mySql_query($sql);
		}

		// TEMP át kell konvertálni az oszlopokakt FLOAT-ról DECIMAL-ra, mert kiegészíti szeméttel a számokat a MySQL
		$sql = "ALTER TABLE _temp_flux MODIFY x_axis DECIMAL(10,5)";
		f_mySql_query($sql);
		$sql = "ALTER TABLE _temp_flux MODIFY y_axis DECIMAL(10,5)";
		f_mySql_query($sql);
		$sql = "ALTER TABLE _temp_flux MODIFY z_axis DECIMAL(10,5)";
		f_mySql_query($sql);

		$x_array = array();
		$y_array = array();
		$z_array = array();

		//lekérjük az X tengely koordinátáit
		$sql = "SELECT x_axis FROM _temp_flux GROUP BY x_axis";
		$response = f_mySql_query($sql);

		//lementjük az x_array ba
		$idx = 0;
		while($row = mysqli_fetch_array($response)){

			if ($row == null){
				echo "X_NULL";
			}
			$x_array[$idx] = $row['x_axis'];
			$idx = $idx+1;
		}
```

```
//lekérjük az Y tengely koordinátáit
$sql = "SELECT y_axis FROM _temp_flux GROUP BY y_axis";
$response = f_mySql_query($sql);

//lementjük az y_array ba
$idx = 0;
while($row = mysqli_fetch_array($response)){

                if ($row == null){
                        echo "Y_NULL";
                }

                $y_array[$idx] = $row['y_axis'];
                $idx = $idx+1;
}

$idx = 0;
foreach($x_array as $x_value){
        foreach($y_array as $y_value){
                usleep(100);
                $sql = "SELECT sum(z_axis) FROM  _temp_flux WHERE (x_axis =".$x_value. ")
AND (y_axis =".$y_value.");";
                $response = f_mySql_query($sql);

                if ($response == null){
                        $row[0] = 0;
                }
                else{
                        $row = mysqli_fetch_array($response);
                }

                if($row[0] == null){
                        $z_array[$idx] = 0;
                }
                else{
                        $z_array[$idx] = $row[0];
                }
```

```
                    $idx = $idx + 1;
                }
                //echo"</br> DONE";
            }

            echo json_encode(array('x' => $x_array, 'y' => $y_array, 'z' => $z_array));

            //A végén eldobjuk a táblát
            $sql = "DROP TABLE IF EXISTS _temp_flux";
            f_mySql_query($sql);
        }

?>
```

5.8.3. DBHandler.php

```
<?php

#########################################################
# MySQL adatbázis kapcsolatkezelő és lekérdező szkript#
#########################################################


   $DATABASE_HOST = "localhost";
   $DATABASE_USER = "root";
   $DATABASE_PW = "";
   $mysql_DB = "measurements_db";

        $ServerFolderError = "c:/xampp/htdocs/WebServ/ERROR/";
        $ServerErrorLogFileName = "error_log.txt";

        function f_error_log($p_Str,$p_filePath){
      $myfile = fopen($p_filePath, "a");
      fwrite($myfile, ($p_Str."  -  " . date("Y-m-d H:i:s")."\n"));
      fclose($myfile);
   }


$mysqli_connection=@mysqli_connect($DATABASE_HOST,$DATABASE_USER,$DATABASE_PW,$mysql_DB);

   if (mysqli_connect_errno()) {
```

```
    //Nem sikerült a MySQL kiszolgálóval felvennei a kapcsolatot;
    f_error_log("Connecting to MySQL database is failed" , $ServerFolderError.$ServerErrorLogFileName);
  }
  else{
    mysqli_set_charset($mysqli_connection,'utf8');
    mysqli_query($mysqli_connection,"SET collation_connection = utf8_hungarian_ci");
  }
  // MySQL Lekérdező függvény
  function f_mySql_query($sql){
    global $mysqli_connection;

    $response = @mysqli_query($mysqli_connection,$sql);

    if (!$response){
      // 'Nem sikerült a lekérdezés';
      return null;
    }
    else{
      return $response;
    }
  }
?>
```

5.8.4. IncomHandler.php

```
<?php
###################################
# Bejövő mérési adatokat feltöltő szkript#
###################################

include_once('..\WebServ\DBHandrel.php');

  $directory = '..\WebServ\INCOMING';
  $allTXTname =  array();
  $allTHPEname = array();
  $allFLUXname = array();
  $ServerFolder = "c:/xampp/htdocs/WebServ/INCOMING/";
      $ServerFolderDone = "c:/xampp/htdocs/WebServ/DONE/";
      $ServerFolderError = "c:/xampp/htdocs/WebServ/ERROR/";
```

```
        $ServerErrorLogFileName = "error_log.txt";

        function f_error_log($p_Str,$p_filePath){
    $myfile = fopen($p_filePath, "a");
    fwrite($myfile, ($p_Str."  -  " . date("Y-m-d H:i:s")."\n"));
    fclose($myfile);
}

function f_move_file($p_from,$p_to){
    rename($p_from,$p_to);
}

#Megkeresük a mappában az összes file nevet
$scanned_directory = array_diff(scandir($directory), array('..', '.'));

$idx = 0;
$id = 0;
$i = 0;

#megkeressük az összes .txt-t tartalamzót, majd szortírozzuk őket
foreach($scanned_directory as $value){
    if (strpos($value,".txt") != false ){

        $allTXTname[$idx] = $value;
        $idx = $idx +1;

        #Ha THPE-ről van szó
        if (strpos($value,"_THPE.txt") != false ){
            $allTHPEname[$id] = $value;
            $id = $id +1;
        }
        #Ha flux ról van szó
        if (strpos($value,"_FLUX.txt") != false ){
            $allFLUXname[$id] = $value;
            $i = $i +1;
```

```
        }

      }
   }
   ###################
   #THPE feldolgozása#
   ###################
   foreach($allTHPEname as $THPEfile){
                        $TempRPIname = "";
              #kivesszük a _THPE.txt -részt
      $TempRPIname = trim($THPEfile, "_THPE.txt");

      #megvizsgáljuk, hogy a THPE_lookup ban szerepel-e ez az RPI
      $sql = "SELECT 1 FROM thpe_lookup WHERE RPi_name ='".$TempRPIname."';";
      $response = f_mySql_query($sql);

      #még nem szerepel az rpi listában
      if (mysqli_fetch_array($response) ==  false) {
         #listához való hozzáadás
         $sql = "INSERT INTO thpe_lookup VALUES ('".$TempRPIname."');";

         $response = f_mySql_query($sql);
         usleep(500);

         if ($response == null){
            f_error_log("RPi name was not in the Rpi_lookup table, but could not insert it - sql: " .$sql ,
$ServerFolderError.$ServerErrorLogFileName);
         }
         else{
            #tábla kreálás CREATE TABLE
            $sql = "CREATE TABLE ".$TempRPIname."_thpe_data (efficiency   FLOAT NOT NULL,
                                                                  Amb_H          FLOAT NOT NULL,
                                                                  Inn_H  FLOAT NOT NULL,
                                                                  Amb_T          FLOAT NOT NULL,
                                                                  Inn_T  FLOAT NOT NULL,
```

```
                                                    Amb_P           FLOAT NOT NULL,
                                                    Inn_P  FLOAT NOT NULL,
                                                    Sample_Date datetime NOT NULL PRIMARY
KEY);";
            $response = f_mySql_query($sql);
            usleep(500);

            if ($response == null){
               f_error_log("Cannot create thpe dara table for rpi  " .$sql,$ServerFolderError.$ServerErrorLogFileName);
            }
            else{
               #file beolvasás
               $sql = "LOAD DATA INFILE '".$ServerFolder.$THPEfile."' INTO TABLE
".$TempRPIname."_thpe_data;";
               $response = f_mySql_query($sql);
               usleep(500);

               if($response == null){
                  f_error_log("Could not load data from file  " .$sql);
               }
                                                    else{
                                                            #Minden ok, volt, file átmozgatása a DONE mappába
                                                            f_move_file($ServerFolder.$THPEfile,
$ServerFolderDone.$THPEfile);
                                                    }
            }
         }
      }
      else{
         #szerepel az rpi listában
         #file beolvasás a táblájába
         $sql = "LOAD DATA INFILE '".$ServerFolder.$THPEfile."' INTO TABLE ".$TempRPIname."_thpe_data;";
         $response = f_mySql_query($sql);
         usleep(500);

         if($response == null){
```

```
        f_error_log("Could not load data from file  " .$sql,$ServerFolderError.$ServerErrorLogFileName);
    }
                    else{
                        #Minden ok, volt, file átmozgatása a DONE mappába
                        f_move_file($ServerFolder.$THPEfile, $ServerFolderDone.$THPEfile);
                    }
  }
}
      ###################
#FLUX feldolgozása#
###################
$TempRPIname = "";
foreach($allFLUXname as $FLUXfile){
            #File név felbontása  Rpi névre és dátumra
            $fileDate = "";
            $fileRpi = "";
            #dátum formázása a LookUp table-höz és az adat táblanévhez
            $dateForLookUpTable = "";
            $dateForDataTable ="";

            $fileRpi = trim($FLUXfile, "_FLUX.txt");
            $tmpArry = explode("___",$fileRpi);
            $fileRpi = $tmpArry[0];
            $fileDate = $tmpArry[1];

            #Lookup table-hez dátum formátum előállítása: 2015-01-04 12:34:55
            $dateForLookUpTable = str_replace("__"," ",$fileDate);
            $dateForLookUpTable = str_replace("_","-",$dateForLookUpTable);
            $dateForLookUpTable = str_replace("x",":",$dateForLookUpTable);
            #Tábla névhez szükséges név előállítása 2015_01_04_12_34_55
            $dateForDataTable = str_replace("__","_",$fileDate);
            $dateForDataTable = str_replace("x","_",$dateForDataTable);

            #megvizsgáljuk, hogy a FLUX_lookup ban szerepel-e ez az RPI
  $sql = "SELECT 1 FROM flux_lookup WHERE RPi_name ='".$fileRpi."' AND time_section
```

```
='".$dateForLookUpTable."';";
        $response = f_mySql_query($sql);

        #még nem szerepel az rpi listában
        if (mysqli_fetch_array($response) ==  false) {
            #listához való hozzáadás
            $sql = "INSERT INTO flux_lookup VALUES ('".$fileRpi."','".$dateForLookUpTable."');";
            $response = f_mySql_query($sql);
            usleep(500);

            if ($response == null){
                f_error_log("RPi name was not in the flux_lookup table, but could not insert it - sql: "
.$sql,$ServerFolderError.$ServerErrorLogFileName);
            }
            else{
                #tábla kreálás CREATE TABLE
                $sql = "CREATE TABLE ".$fileRpi."_".$dateForDataTable." (x_axis FLOAT NOT NULL,

                                                    y_axis FLOAT NOT NULL,

                                                z_axis FLOAT NOT NULL,

                                                id INT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY );";
                $response = f_mySql_query($sql);
                usleep(500);

                                    if ($response == null){
                    f_error_log("Cannot create flux data table for rpi  " .$sql,$ServerFolderError.$ServerErrorLogFileName);
                }
                else{
                    #file beolvasás
                    $sql = "LOAD DATA INFILE '".$ServerFolder.$FLUXfile."' INTO TABLE
".$fileRpi."_".$dateForDataTable.";";
                    $response = f_mySql_query($sql);
                    usleep(500);
```

```
                        if($response == null){
                f_error_log("Could not load data from file  " .$sql,$ServerFolderError.$ServerErrorLogFileName);
            }
                                    else{
                                        #Minden ok, volt, file átmozgatása a DONE mappába
                                        f_move_file($ServerFolder.$FLUXfile,
$ServerFolderDone.$FLUXfile);
                                    }
          }
        }
      }
      else{
        #szerepel az rpi listában és ez hiba
                        #Hiba, bejegyzés és átmozgatás
                        f_error_log("Rpi: ".$fileRpi." whit date: ".$dateForDataTable." already in fulx_lookup table,
this file is
                                    going to move to ERORR folder. File name: "
.$FLUXfile,$ServerFolderError.$ServerErrorLogFileName);
                        f_move_file($ServerFolder.$FLUXfile, $ServerFolderError.$FLUXfile);

      }
   }

?>
```

5.8.5. PlotChart.php

```
<?php
################################
# Chart megjelenítést vezérlő szkript#
################################

$TYPE =$_REQUEST['TYPE'];

   if($TYPE == 1){
      echo "<div class=\"center\" id=\"container\" style=\"min-width: 310px;height:400px;border:1px solid
blueq\"></div></br>";
   }
```

```
    elseif($TYPE == 2){
        echo "<div class=\"center\" id=\"container_flux\" style=\"min-width: 1000px;height:1000px;border:1px solid blueq\"></div></br>";
    }


?>
```