



M Ű E G Y E T E M 1 7 8 2

**Budapesti Műszaki és Gazdaságtudományi Egyetem**  
Villamosmérnöki és Informatikai Kar

Krainhoffer Nóra, Szőri Bence

# **GOMBA TERMESZTŐ KONTÉNER AUTOMATIZÁLÁSA**

Tudományos Diákköri Konferencia Dolgozat

KONZULENS

**Dr. Szabó Sándor**

BUDAPEST, 2023

# 1. Tartalomjegyzék

1.	Tartalomjegyzék.....	2
2.	Kivonat.....	3
3.	Abstract.....	4
4.	Bevezetés .....	5
5.	Gombafogyasztási szokások .....	10
6.	Jelenlévő technológiák (Holland házak).....	11
7.	Eszközök kiválasztása és programozása Arduino IDE segítségével.....	13
7.1.	Tervek .....	13
7.2.	I2C kommunikáció .....	14
7.3.	Eszközök kiválasztása .....	15
7.4.	Arduino IDE .....	18
7.5.	Programkódok .....	18
8.	Kommunikáció az eszközök között .....	21
8.1.	M2M .....	22
8.2.	MQTT protokoll .....	22
9.	Adatgyűjtés a helyszínen .....	28
9.1.	Eredmények .....	28
9.2.	Folytatás.....	28
10.	Irodalomjegyzék .....	30

## 2. Kivonat

A mezőgazdaságban megnőtt az igény, hogy kevesebb emberi jelenléttel, kevesebb befektetett nyersanyaggal jobb eredményt lehessen elérni a termelésben. Érzékelők, műholdak, beágyazott rendszerek segítségével hatékonyan, pontosan, a megfelelő helyen és időben - ez a lényege az okos farming-nak.

Dolgozatunk témája egy gombatermesztő konténer automatizálása szenzorok és mikrokontrollerek segítségével. Jelenleg a gombákat általában nagy gombafarmokon termesztik. Itt hatalmas csarnokokban, hosszú polcrendszereken nevelik a gombákat. Ez segít az ideális környezeti feltételek megtartásában a legkevesebb energiabefektetés mellett. A hátránya a nagy mennyiségű előállításnak az, hogy meg kell oldani a termény elszállítását. A gombákat magas víz és fehérjetartalmuk miatt nem lehet hosszú ideig tárolni, így a távoli exportálás sem megoldás. Erre a problémára kínál megoldást az automatizált gombatermesztő konténer.

Jelenleg a piacon kétféle módszer létezik helyszíni gombatermesztésre. Az egyik a korábban említett, óriási méretű, termesztősátras megoldás, amely igen helyigényes és drága, továbbá fennáll a termés elszállításának nehézsége. A másik lehetőség pedig azoknak alkalmas, akik kis mennyiségű gombát szeretnének lakásukban termesztetni egy gombnyomással. Ugyan ez kevésbé drága, de még ez is egy olyan megoldás, aminek a fenntartására igen költséges.

Mi arra kerestünk megoldást, hogy legyen még egy lehetőség, amely nem igényel a konténeren kívül egyéb nagyobb beruházást, egyszerű az elektronikája, távolról elérhető, vezérelhető a munkaerő minimalizálása érdekében, mobilis, könnyen bővíthető és ellát akár egy kisebb falunyi embert a benne termesztett gomba.

A konténerben való automatizálást mikrokontrollerekkel valósítottuk meg. Rendszerünk alapját egy Raspberry Pi eszköz adta. Ezzel valósítottuk meg a távoli elérést és az ESP8266 mikrokontrollerekről való adatgyűjtést. Az esp-vel a konténerben három helyen tudtuk monitorozni a gombák számára fontos körülményeket. Ezek a körülmények pedig a hőmérséklet, páratartalom és szén-dioxid szint. A mikrokontrollerek vezeték nélküli kommunikációja biztosítja a konténeren belüli áthelyezhetőséget valamint a könnyű felszerelést. Ezt a kommunikációt az MQTT (Message Queuing Telemetry Transport) adatkapcsolattal valósítjuk meg. A rendszer távolról elérhető és monitorozható, így nyomonkövethetjük a rendszer működését akár otthonról is.

### **3. Abstract**

In agriculture, there is an increasing demand to achieve better results in production with fewer human presence and less invested raw materials. Through sensors, satellites, and embedded systems, efficiently, accurately, at the right place and time - this is the essence of smart farming.

The topic of our paper is the automation of a mushroom cultivation container using sensors and microcontrollers. Currently, mushrooms are typically grown on large mushroom farms. They are cultivated in vast halls on long shelving systems to maintain ideal environmental conditions with minimal energy input. The downside of large-scale production is the need to solve the transportation of the produce. Due to their high water and protein content, mushrooms cannot be stored for long periods, making distant exportation unfeasible. The solution to this problem is an automated mushroom cultivation container. Such cultivation containers already exist in the industry, but we were looking for a solution that is cost-effective, easily expandable, and mobile.

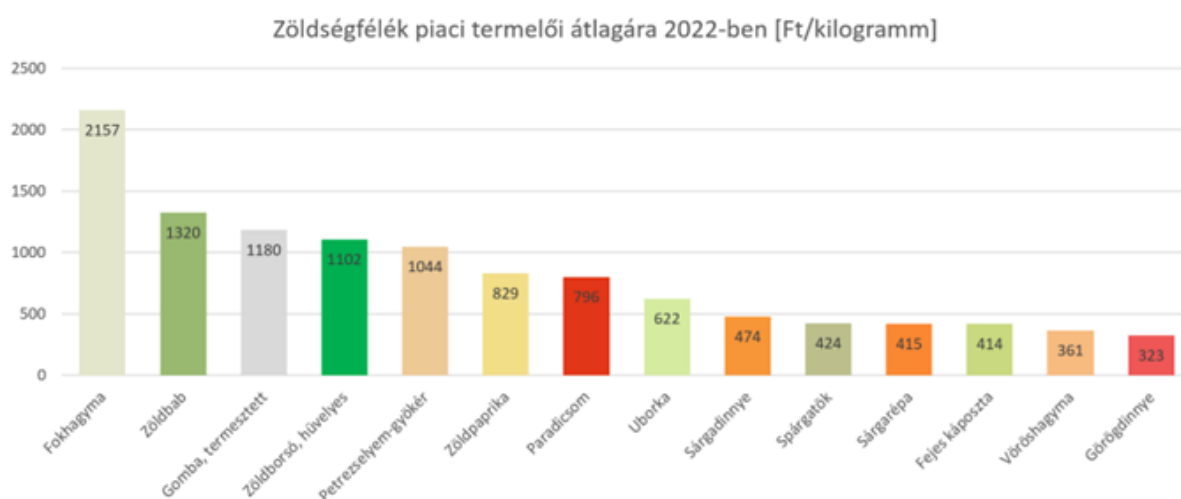
The automation inside the container was implemented using microcontrollers. The foundation of our system was a Raspberry Pi device. This allowed us to achieve remote access and data collection from esp8266 microcontrollers. With the esp, we could monitor the conditions important for the mushrooms at three locations inside the container. These conditions are temperature, humidity, and carbon dioxide levels. The wireless communication of the microcontrollers ensures portability within the container and easy installation. This communication is achieved through the MQTT (Message Queuing Telemetry Transport) data connection. The system is accessible and monitorable remotely, allowing us to track its operation even from home.

## 4. Bevezetés

Az elmúlt években a gombák népszerűsége nemcsak kulináris jellegük miatt nőtt, hanem potenciális környezeti előnyeik miatt is. Fenntartható élelmiszerforrásként minimális erőforrásokat és helyet igényelnek a termesztéshez, ezáltal vonzó lehetőséget jelentenek azoknak, akik környezetbarát alternatívákat keresnek mindennapi étkezésükben.

A Monterey Mushroom (2019) tanulmánya szerint egy font (0.45 kg) csiperkegomba előállításához átlagosan ~6,8 liter vizet igényel (míg a paradicsomhoz ~98 liter víz szükséges), körülbelül 1 kilowattóra energia (ez egy kb. egy bögre kávé főzésének és melegen tartásának megfelelő energia), és ~0,3 kilogramm szén-dioxid kibocsátást eredményez (míg az emberek napi átlagosan 1 kilogramm szén-dioxidot termelnek). Ezek az adatok azt mutatják, hogy a gombatermesztés az egyik legfenntarthatóbb élelmiszertermelési forma közé tartozik az európai viszonylatban.

A népszerűség növekedés egy új problémát vetett fel a termelők körében. A gombákat el is kell szállítani a fogyasztókhöz, ráadásul mihamarabb, mivel a gomba, a magas víz- és fehérjetartalmának köszönhetően kifejezetten gyorsan romló élelmiszer. A hűtőkamionos szállítás a gombatermelés szemléletében nagy költséggel jár, nem is meglepő, hogy tavaly a gombák a harmadik legdrágább „zöldségféle” volt a KSH adatai szerint. [4.1. ábra] (Bár mint tudjuk a gombák se nem növények se nem állatok, így a zöldségfélék közé sem illik sorolni)

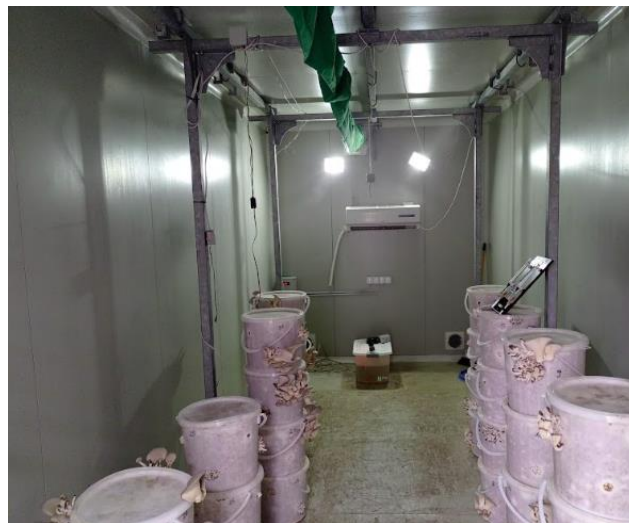


4.1. ábra Zöldségfélék piaci termelői ára 2022-ben [7]

A kertészmérnöki egyetemmel kapcsolatba lépve becsatlakoztunk egy gombatermesztő konténer [4.2. ábra][4.3. ábra] fejlesztésébe, villamosmérnöki oldalról. A projektet 2023 tavaszán kezdtük, ekkor kezdtük el a helyszín felmérését, a gondolatok összeszedését, a rendszer felépítésének megtervezését. A gombakonténer ekkor Soroksáron, a Kertészeti Egyetem tangazdaságában volt elhelyezve, így már a megközelítés is eléggé macerásnak bizonyult. A projekt kezdetén azt mondták, hogy a konténer ott a helyén fog maradni még 1-2 évig biztosan, ezért nyugodtan kezdtünk bele a megvalósításba. Többször voltunk kint a helyszínen felmérni az igényeket, hogy minél megfelelőbb rendszert tudjunk kiépíteni. Felmértük a hálózati lefedettséget, internetelérést, áramhoz jutási lehetőségeinket. Maga a konténer egy a szállítmányozásban legtöbbször használt általános 20 lábás konténer, ami 6 méter hosszú, körülbelül 2,5–2,5 méter széles és magas. Ezt a konténert ellátták a megfelelő szigeteléssel, valamint a fűtéshez és hűtéshez szükséges klímával, egy hidegpárásító berendezéssel, valamint egy extra szellőztetést biztosító csőrendszerrel.



*4.2. ábra A gombatermesztő konténer*



*4.3. ábra A konténer belseje*

Feladatunk a konténerben található körülmények optimalizálása, melyre igen nagy szüksége van a benne termesztett gombáknak. Tapasztaltuk, hogy korábban analóg mérőeszközökkel lehetett vizsgálni a konténer páratartalmát és hőmérsékletét, valamint a párásító és szellőztető berendezés is csak helyszíni jelenléttel volt vezérelhető. A

légkondicionáló, a párasító és a szellőző közül egyik sem működött megfelelően, hiszen a párának 90% körülnek kellett volna lennie, a levegő elhasznált volt és a hőmérséklet jóval alacsonyabb volt a kelleténél majdnem mindig, amikor ott jártunk. [4.4. ábra][4.5. ábra][4.6. ábra]



4.4. ábra Analóg hőmérő és hidegpáramérő eszköz



4.5. ábra A párasító nem termel elég párat



4.6. ábra A légkondicionáló működik, de a szellőztető rendszer nem cseréli ki megfelelően a levegőt

A sok tervezés és kisebb helyszíni mérések után 2023 nyarán a konténert minden eszközével együtt elszállították, így az eredeti elképzeléseinken változtatnunk kellett. Ugyan a

kertészeti egyetem továbbra is biztosított nekünk egy gomba garázst, az bent van a városban az egyetem területén, ahol sokkal nagyobb a hálózati lefedettség, temérdek Wifi hálózat van egyszerre és a vezetékes internetre való feljutás is körülményes az egyetemi hálózat miatt. Ezen okok miatt újra kellett terveznünk időközben a rendszer felépítését. Alapvető tervünk az volt, hogy a helyszínen lévő vezetékes internetet egy routerrel Wifi hálózattá alakítjuk, és ezen a hálózaton tudnak a mikrokontrollerjeink adatot küldeni és fogadni egymás és az internet között is. Ez a költözéssel megvalósíthatatlanná vált ugyanis az egyetemen lévő menedzselt hálózat nem engedélyez külső, nem általuk telepített router bekapcsolását a hálózatba. Azt sikerült megbeszelnünk a rendszergazdákkal, hogy a vezetékes hálózatba bekapcsolhassuk a Raspberry pi eszközünket, ami a fő szerepet ellátja. Ezután bekapcsoltunk a rendszerbe egy saját, a hálózatra nem kötött routert, melyre az eszközeinket csatlakoztatva, azok egymással kommunikálni tudnak, így kikerülve az egyetemi Wifi hálózat szabályait.

Magát a gombatermesztést három fázisra lehet bontani: komposzt előkészítése és gombafonalakkal történő átszövetése, lappangási idő és termőre fordítás, növekedési és betakarítási fázis. Az egész termesztés során mindig a szakasznak megfelelő hőmérsékletet és páratartalmat kell biztosítani, hogy a komposzt és a termőtest ne száradjon ki, valamint, hogy egészséges és homogén legyen a gombahozam. Továbbá figyelni kell a levegőcserére, hiszen a levegő magas szén-dioxid tartalma a termesztés első szakaszában hasznos csupán, a második szakaszában már káros.

A környezeti tényezők pontos szabályozásához először is adatgyűjtésre van szükség. A konténerben a legnagyobb feladatot az adatok mérése, tárolása, továbbítása, valamint feldolgozása jelenti, hiszen számolni kell a helyszín infrastrukturális felszereltségével is. Az internet elérés hiánya, a magas páratartalom, a por és egyéb szennyeződések mind olyan tényezők, amik nehezítik a feladat megoldását. Az adatok megjelenítése webes alapon működik, mely lehetőséget ad a távoli hozzáférésre.

A konténerünk laskagombákra specializálódott, így elsősorban az ezek számára fontos környezeti tényezőket vizsgáljuk.





**4.7. ábra Laskagombák a termesztés második fázisában a termesztőkonténerben**

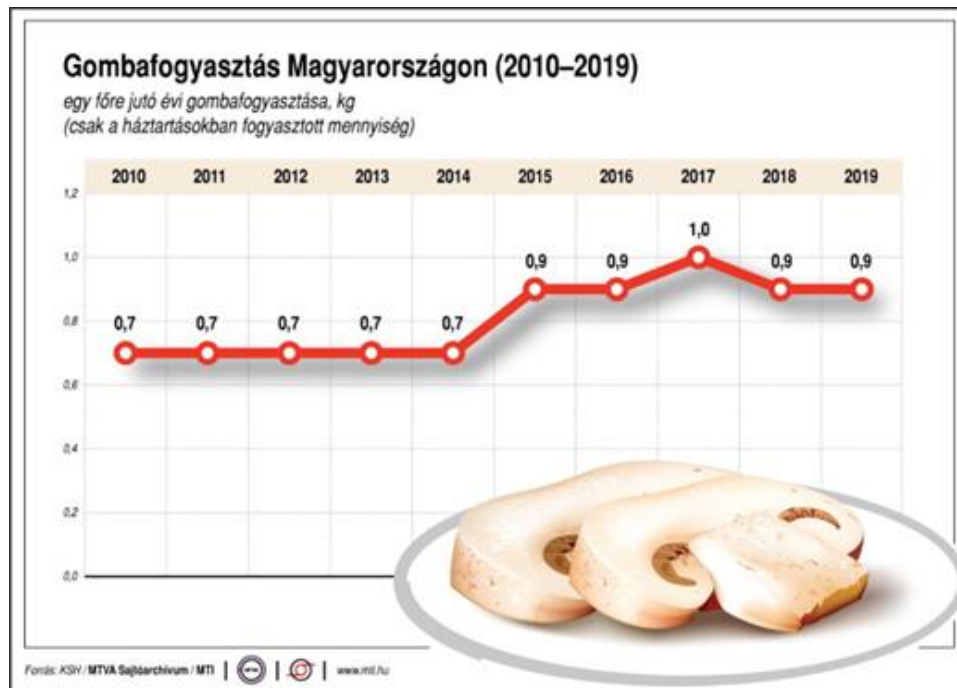
A konténerben három helyen három-három értéket tudunk mérni. Ezek:

- Hőmérséklet
- Páratartalom
- Szén-dioxid

A szenzorokat három darab ESP 8266 mikrokontroller vezérli melyek kommunikálnak egy Raspberry Pi 3-mal. A Pi kezeli a beavatkozókat a begyűjtött adatok alapján, valamint az adatok tárolását és továbbítását végzi. Mivel a konténert felhasználóbarátnak terveztük, hogy az eszközök kihelyezésén kívül semmilyen kábelezésre ne legyen szükség, (energiaellátáson kívül persze) a mikrokontrollerek egymással vezetékmentesen kommunikálnak. Ezen kommunikáció alapját a WiFi hálózat adja.

## 5. Gombafogyasztási szokások

A gombatermesztés Magyarországon egy nehezen kifizetődő vállalkozás, ugyanis a gombafogyasztási szokásaink elég alacsonyak és változatosak. A KSH adatai szerint 2010-ben az egy főre jutó gombafogyasztás 0,7 kg volt [5.1. ábra]. Azóta az elmúlt években növekedésben van a gombafogyasztás, vélhetően a 2020-as években még feljebb fog menni ez a szám a vegetáriánus életmód elterjedésével.

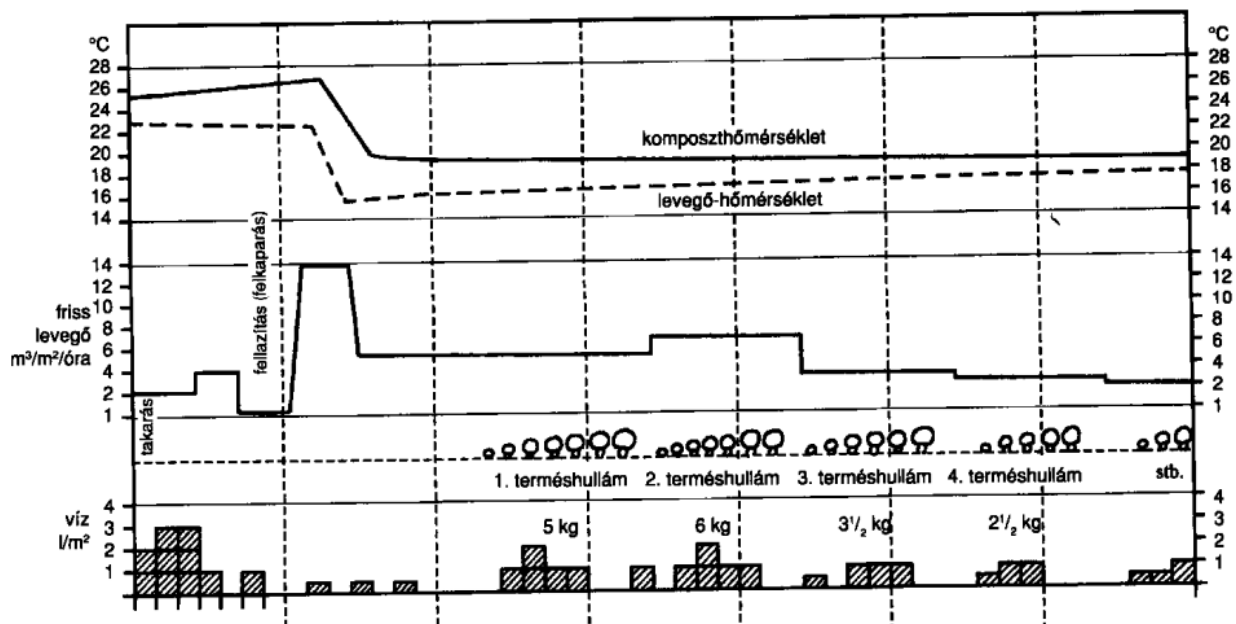


5.1. ábra Gombafogyasztás Magyarországon 2010-2019 [6]

Ám ha megnézzük a régiónkénti lebontást látható, hogy amíg a fővárosban 2010-ben 1,1 kg addig például az Észak-Alföldön csak 0,4 kg az egy főre jutó éves mennyiség. [6]

## 6. Jelenlévő technológiák (Holland házak)

Napjainkban a legelterjedtebb termesztési módszer az úgynevezett holland házas termesztés. A holland házak kifejezetten a gombatermesztésre lettek specializálva. Az egész csarnokon keresztülfutó rozsdamentes polcsorok között éppen csak annyi hely van, hogy a munkás beférjen. Kezdetben a komposztot nagy zsákokba téve elkezdik átszövetni azokat. Ez a folyamat 22-24°C os levegő hőmérsékleten és 95-98 % os relatív páratartalomon történik. Az átszövetés során a csíraszemek körül fehér micéliumszövedék alakul ki, mely 14-18 nap alatt beszövi a teljes komposztot. A polcsorokra kerül a beszövetett komposzt több centiméter vastagon majd letakarják 5-6 centiméternyi mészkőpor és tőzeg keverékével. Ezután jön a várakozási időszak. Ebben az időszakban figyelni kell folyamatosan a megfelelő páratartalom és hőmérséklet megtartására. A takarás utáni 10-15. napon kezdődik el az apró termőtestek megjelenése. Ekkor a levegő hőmérsékletét lecsökkentik 16-18°C ra és a relatív páratartalmat is 92% körülire. Ekkor már fontos a rendszeres szellőztetés is, nem szabad, hogy a gombák megfulladjanak, hiszen ők is mint az állatok oxigént lélegeznek be és szén-dioxidot ki. A gombák átlagosan 7-10 nap alatt nőnek meg, és egy adag komposztról 3 terméshullámot szednek le, bár átlagosan 5-6 hullám is teremne, viszont ezek már lényegesen kevesebbet hoznak. A 3 hullám alatt 100kg komposztról 22-34kg gomba szedhető.



6.1. ábra A korszerű holland típusú csiperketermesztés folyamata [1]



**6.2. ábra Holland típusú gombatermesztő ház [2]**

A gombatermesztés ezen formája a leghatékonyabb napjainkban, ám egy ilyen ház felépítése és beüzemelése több hónapot és több tízmillió forint beruházást vesz igénybe. Az iparban megjelentek már a miénkhez hasonló konténeres megoldások is. A Freight Farms vállalat például ilyen konténereket árul. Ezek nem specializálódtak külön növénycsoportokra, viszont gombát nem lehet bennük termesztani, ugyanis itt hidropon módszerrel termesztnek. Ez sajnos a gombáknak nem megfelelő termesztési módszer. Ezen felül az ő konténerük is 23 millió forintba kerül megvásárláskor, majd erre jön még az évi több milliós fenntartási költség.

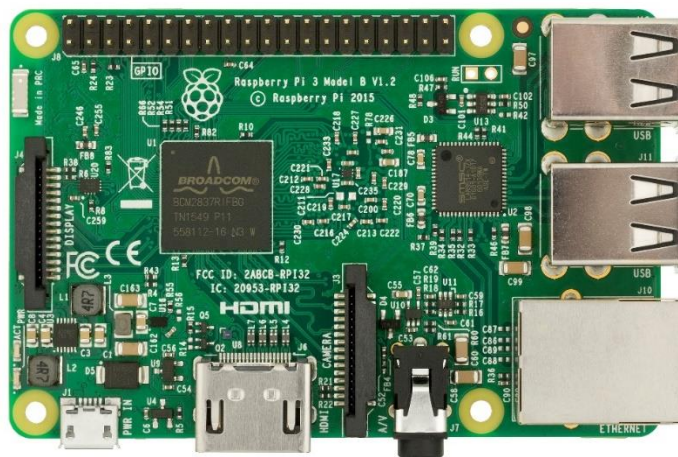
Egy másik konténeres megvalósítás a magyar fejlesztésű mobilgomba. Ez a miénkhez hasonló elven működő gombakonténer, ám hosszas kutatás után se találtunk több információt az ő rendszerükről néhány 2016-os cikknél, amikből nem derül ki sajnos a működési elvük.

# 7. Eszközök kiválasztása és programozása Arduino IDE segítségével

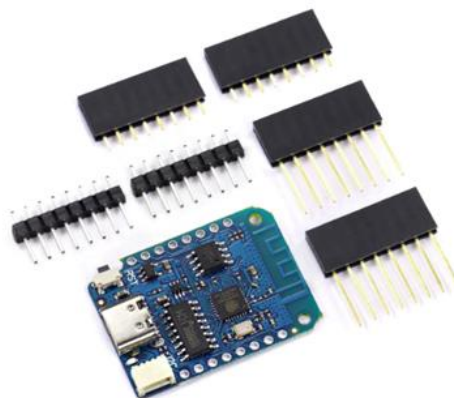
## 7.1. Tervek

A felhasznált szenzorok és mikrokontrollerek kiválasztása elsősorban az alapján történt, hogy mi állt már rendelkezésünkre, majd másodlagosan arra törekedtünk, hogy minél olcsóbban ki tudjunk alakítani egy precíziós környezetet.

Amikor elkezdtük a projektet, rendelkezésünkre állt egy Raspberry Pi 3 modul [7.1. ábra], valamint kettő ESP8266 wemos d1 mini mikrokontroller [7.2. ábra]. A későbbiekben bővült az eszköztárunk még egy ESP8266-tal, amelyet a konzulensünk biztosított. Ezekhez rendeltünk még páratartalom/hőmérséklet, valamint levegőminőség szenzorokat. Az ESP-k és a szenzorok is rendelkeznek I2C (Inter-Integrated Circuit) interfésszel, így ezt használtuk az adatok átvitelére.



7.1. ábra Raspberry Pi 3 modul [11]



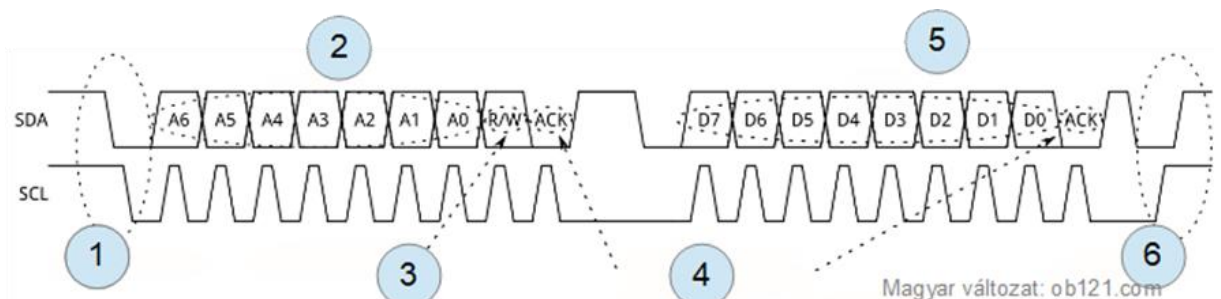
7.2. ábra Esp8266 wemos d1 mini [3]

## 7.2. I2C kommunikáció

Az I2C kapcsolatról azt érdemes tudni, hogy egy busz alapú protokoll, ami két vezeték igényel. Az egyiken megy a sorosított adat (Serial Data, SDA) a másikon pedig a sorosított órajel (Serial Clock, SCL). Ezek közvetítik az információt a buszhoz csatlakoztatott eszközök között.

Minden eszközt (legyen az mikrovezérlő, LCD meghajtó, memória vagy szenzor) általában 7, 10 vagy 16 bites egyedi címmel azonosítanak és az eszköz funkciójától függően működhet adatküldőként vagy adatfogadóként. Például egy szenzor csak adatküldő lehet, míg egy memória képes adatot fogadni és küldeni is. Az adatküldők és adatfogadók mellett az eszközöket vezérlőként vagy céleszközként is értelmezhetjük, amikor adatátvitelt végeznek. Egy vezérlő az az eszköz, amely elindít egy adatátvitelt a buszon, és generálja az órajelet ennek a továbbításnak az engedélyezéséhez. Ebben az időben minden címezett eszközt céleszközként kezelnek. A mi esetünkben az ESP mikrokontroller a vezérlő és a szenzorok a céleszközők.

Az adatátvitelt a következő ábra szemlélteti:



7.3. ábra Adatátvitel az I2C buszon

1. **Start:** Induláskor a master először az SDA-t, majd az SCL-t is alacsonyra húzza. Ez jelzés az összes hálózati résztvevő felé a forgalmazás indítására. Több jelenlevő master esetén az kommunikálhat, amelyik először húzza le az SDA-t. Összeakadás esetén a műveletet ismételni kell.
2. **Címkeret:** A 7 bites címzés esetén ez a tartalom megy ki először. Az MSB az első bit.
3. **R/W:** A nyolcadik bit azt jelzi, hogy olvasási vagy írási művelet következik. (1: olvasás, 0: írás)
4. **NACK / ACK:** A 9. bit Ennél a bitnél a master a címzett egység „reakcióját” várja. Ha az megkapta a telegram eddigi tartalmát, a 9. óraimpulzus előtt le kell húznia az SDA vonalat.



Ha ezt nem teszi meg, a master arra fog következtetni, hogy az egység nem érhető el, folytatja a forgalmazást más egységekkel.

5. **Adatkeret fogadása:** Az adatkeret hossza a protokoll által nincs szabályozva. Sok esetben a slave-ek, ha nagyobb adatmennyiséget kell forgalmazniuk, az egymást követő telegramokban más-más regiszterek tartalmát küldik el.
6. **STOP:** Először az SCL, majd az SDA is magas állapotba lesz „eresztve” a master oldaláról. [5]

Ez a fajta kommunikáció azért előnyös az esetünkben, mert így tetszőlegesen bővíthetők a mérőállomások szenzorokkal. Ha a jövőben esetleg valamilyen más érték monitorozására is szükség lenne, akkor egy I<sup>2</sup>C képes szenzort a többi szenzorral párhuzamosan kötve könnyen meg lehet valósítani. Kábelezés szempontjából is előnyös, a magas páratartalom miatt könnyen korrodálódnak a kábelek, ezért érdemes minél kevesebbet használni, így is csökkentve a meghibásodás valószínűségét hosszútávon.

### 7.3. Eszközök kiválasztása

Mikrokontrollerek közül választhattunk volna Arduino-t vagy esp32-t is, de az esp8266 wemos d1 mini egy tökéletes köztes megoldásnak bizonyult, mind teljesítményben, mind pedig árban. Az Arduino-val ellentétben az esp8266-ot lehet WIFI-n keresztül vezérelni és monitorozni, viszont nem tud Bluetooth-szal kommunikálni, mint az esp32. [7.4. ábra]

Mikrokontroller	ESP8266 WeMos D1 Mini	Arduino UNO R3	ESP32 NodeMCU
Wifi	Van	Nincs	Van
Bluetooth	Nincs	Nincs	Van
Ár	\$6	\$22	\$11

7.4. ábra Mikrokontrollerek összehasonlítása [8]

Három darab esp8266-os mikrokontroller által vezérelt szenzorcsoportot alakítottunk ki, mindegyik esp-re egy hőmérséklet-pára, valamint egy levegőminőség szenzort kötöttünk.

Több fajta hőmérséklet/páratartalom mérő szenzor is található a piacon, ezek közül a legpontosabbnak és legelérhetőbbnek az SHT-kat találtuk. A hőmérsékletet és a párat egy SHT szenzor méri, aminek két fajtáját használtuk fel. Az egyik az SHT30 szenzor [7.7. ábra], amelyet

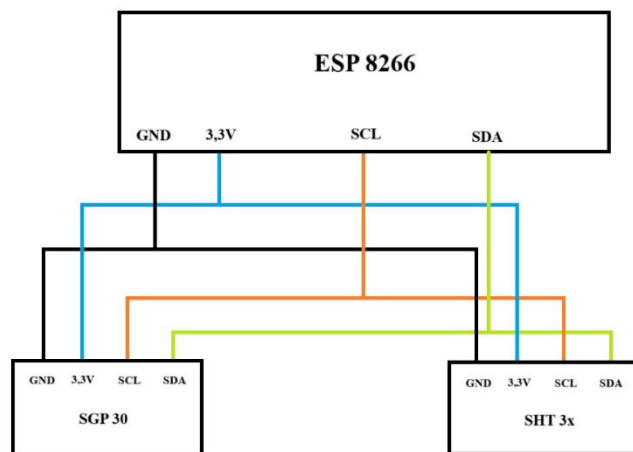
shield módszerrel [7.6. ábra] kötöttük rá az esp-re, a másik pedig az SHT szenzorcsalád egy újabb típusa, az SHT3x. [7.8. ábra]

Két szenzor összehasonlítása [13][14]:

	SHT30	SHT3x
Hőmérséklet (pontosság, intervallum)	$\pm 0.3^{\circ}\text{C}$ , $-40-125^{\circ}\text{C}$	$\pm 0.3^{\circ}\text{C}$ , $-40-90^{\circ}\text{C}$
Pára mérésének pontossága	$\pm 3\%$	$\pm 2\%$

A levegőminőség szenzor egy SGP30-as nagy pontosságú légszennyezettség-érzékelő. [7.9. ábra] Méri a levegő VOC, azaz volatile organic compounds, illékony szerves vegyület összetételét, valamint a szén-dioxid szintet. Kompakt mérete, alacsony energiafogyasztása és a pontos VOC-mérései révén kiváló választás a légszennyezettség monitorozására otthoni, irodai és ipari környezetben egyaránt. Mi a feladatban nem használtuk a VOC mérő képességét az eszköznek, hiszen a gombák esetén csak a szén-dioxid szint mérése fontos. [12]

Az ESP-ket a következő módon kötöttük össze a szenzorokkal:



7.5. ábra Egy kialakított mérőállomás kapcsolási rajza



Az eszközök:



7.6. ábra SHT30 szenzor és esp8266 shield módszerrel összekötve



7.7. ábra SHT30 szenzor [13]



7.8. ábra SHT3x szenzor [14]



7.9. ábra SGP30 szenzor [12]

## 7.4. Arduino IDE

Az Arduino IDE, azaz Integrated Development Environment, egy nyílt forráskódú fejlesztői környezet, melyet beágyazott rendszerek és elektronikai eszközök programozására használnak. Mi C++ nyelven programoztuk fel az ESP-eket, melyekhez szükség volt több könyvtár letöltéséhez is. A platform támogatja, hogy saját könyvtárakat írjanak a felhasználók a különböző eszközökhöz, valamint vannak beépített könyvtárjai is, amelyek használhatók számos szenzorhoz és modulhoz.

Telepíteni kellett a környezeten belül a Board Manager segítségével a megfelelő board-ot, ami a mi esetünkben az ESP8266 volt. Ezáltal tudjuk használni a szükséges függvényeit, például a helyi wifi hálózathoz való csatlakozáshoz szükséges metódusokat. GitHub-on megtalálható számos olyan könyvtár, amit a felhasználói közösség tagjai írtak és fejlesztettek. Ezek közül kellett kiválasztanunk és integrálnunk a projektbe azokat, amelyek az SHT és SGP szenzorainkhoz a legmegfelelőbbek.

## 7.5. Programkódok

A három ESP-re azonos c++ kódot töltöttünk, egyedül az MQTT-hez szükséges topic-ok [8.2] elnevezését kellett megváltoztatni, valamint az SHT3x-es szenzornál is más könyvtárat kellett hozzáadni a fejlesztői környezethez.

Az esp-k MQTT kliensekként működnek, amik a következő témákra teszik közzé az adatokat:

```
gomba/esp[szám]/humidity  
gomba/esp[szám]/temperature  
gomba/esp[szám]/co2
```

Az esp-k számozása 1-től 3-ig van, mindegyik esp mindegyik témára közzéteszi az általa gyűjtött adatot. A következőkben az 1-es számú esp8266 kódjának fontosabb részeit fogjuk bemutatni. Az MQTT-hez kapcsolódó változók a kódban a következő módon vannak létrehozva:

```
// MQTT  
const char* mqtt_server = "192.168.43.224"; // IP of the MQTT broker  
const char* humidity_topic = "gomba/esp1/humidity";  
const char* temperature_topic = "gomba/esp1/temperature";  
const char* co2_topic = "gomba/esp1/co2";
```

Továbbá meg kellett adni, hogy melyik porton érhető el az MQTT broker:

```
// 1883 is the listener port for the Broker
PubSubClient client(mqtt_server, 1883, wifiClient);
```

Az 1883-as port van fenntartva az MQTT-nek. Ezen titkosítatlanul mennek az üzenetek, míg a 8883-as porton titkosítva.

Energiatakarékossági szempontból megoldható lenne, hogy az eszköz lecsatlakozzon bizonyos időközönként az MQTT brokerről, de mi ezt nem használtuk, mert nem fogyaszt annyi energiát egyik esp sem.

A következő két változó a két szenzor, amit egy esp kezel:

```
SHT3x mySHT;
SGP30 mySGP;
```

Ezeknek az adatait a következő módon kértük le inicializálás után:

Hőmérséklet és pára szenzor:

Levegőminőség szenzor:

```
mySHT.UpdateData();
Serial.print("Temperature: ");
float t = mySHT.GetTemperature();
Serial.print(t);
Serial.write("\xC2\xB0"); //The Degree symbol
Serial.println("C");
Serial.print("Humidity: ");
float h = mySHT.GetRelHumidity();
Serial.print(h);
Serial.println("%");
```

```
mySGP.measureAirQuality();
float co2 = mySGP.CO2;
Serial.print(co2);
```

A lekért adatokat át kellett alakítani String formátumúvá, mert az MQTT csak szöveget tud kezelni. Ehhez a következő konverziókra volt szükség:

```
// MQTT can only transmit strings
String hs=String((float)h);
String ts=String((float)t);
String cs=String((float)co2);
```

Ezután már csak továbbítani kellett a megadott témákra a gyűjtött adatot, valamint ellenőrizni, hogy csatlakozott-e a kliens a brokerhez.

### Hőmérséklet kódja:

```
if (client.publish(temperature_topic, ts.c_str())) {
  Serial.println("Temperature sent!");
}
else {
  Serial.println("Temperature failed to send. Reconnecting to MQTT Broker and trying again");
  if (!client.connected()) {
    connect_MQTT();
  }
  delay(10); // This delay ensures that client.publish doesn't clash with the client.connect call
  if (client.publish(temperature_topic, ts.c_str())) {
    Serial.println("Temperature sent!");
  }
}
```

### Páratartalom kódja:

```
if (client.publish(humidity_topic, hs.c_str())) {
  Serial.println("Humidity sent!");
}
else {
  Serial.println("Humidity failed to send. Reconnecting to MQTT Broker and trying again");
  if (!client.connected()) {
    connect_MQTT();
  }
  delay(10); // This delay ensures that client.publish doesn't clash with the client.connect call
  if (client.publish(humidity_topic, hs.c_str())) {
    Serial.println("Humidity sent!");
  }
}
```

### CO2 kódja:

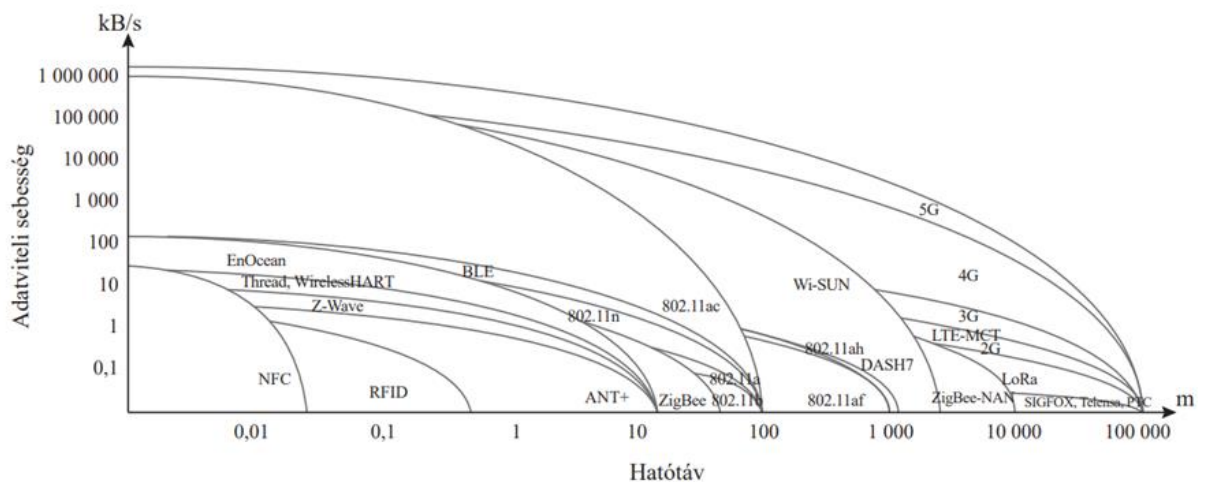
```
if (client.publish(co2_topic, cs.c_str())) {
  Serial.println("CO2 sent!");
}
else {
  Serial.println("CO2 failed to send. Reconnecting to MQTT Broker and trying again");
  if (!client.connected()) {
    connect_MQTT();
  }
  delay(10); // This delay ensures that client.publish doesn't clash with the client.connect call
  if (client.publish(co2_topic, cs.c_str())) {
    Serial.println("CO2 sent!");
  }
}
```

## 8. Kommunikáció az eszközök között

A vezeték nélküli kommunikációs technológiák kiválasztásánál három fő szempontot érdemes figyelembe venni.

Az első, hogy mekkora a rendszer maximális adatátvesztő képessége (throughput). Ezt fontos tudni ugyanis ez az érték befolyásolja a csatorna gyorsaságát. Értéke pár tíz kB/s-tól egészen pár Gbit/s-ig terjedhet a mai technológiákkal. Nagy áteresztőképesség elsősorban videó vagy hangátvitelhez szükséges, esetünkben az M2M (Machine to Machine) kommunikációban a szenzoros eredmények átvitele nem jelent túl nagy mennyiséget.

Másodikként a maximális kapcsolódási távolság, ami fontos szerepet játszik a választásban. Itt is nagyon tág a skála, az érintkező, pár centiméteres maximum távolságtól (pl: NFC) egészen a több kilométeres hatótávolságig (pl:LoRa). Ennek eldöntésekor figyelembe kell vennünk a területet, amit le szeretnénk fedni. Jelen esetben egy egészen behatárolható területünk van, maga a konténer, ami 6 méter hosszú. Ennél van eggyel nagyobb méretű konténer is de ez se haladja meg a 13 métert.



8.1. ábra IoT-ben használt vezeték nélküli adatátviteli protokollok összehasonítása [15]

Végül fontos még az energiafelhasználás. Egy akkumulátorról üzemeltetett rendszer esetében fontos, hogy ott spóroljunk az energiával, ahol tudunk. Ez esetünkben elhanyagolható mivel a beavatkozó eszközök energiafogyasztása lényegesen több mint bármelyik rövid vagy közepes távú csatornának. Ráadásul a beavatkozók miatt elengedhetetlen a hálózati áram így az akkumulátoros üzem szóba se jöhet.

Mindezeket figyelembe véve választásunk az IEEE 802.11n WIFI szabványra esett. Ez napjainkban az egyik legelterjedtebb wifi szabvány így kompatibilitási problémákba se

ütközünk későbbi bővítés során. Hatótávolsága kitűnő a konténerben, a routert bárhová elhelyezve tökéletes lefedettséget kapunk. Mind az ESP-k mind pedig a Raspberry Pi eszköz beépített WIFI modult tartalmaz, így külön modulokra sincs szükség. A szabvány átlagos átviteli sebessége 100 Mb/s mely bőven sok is a mi minimális adatmennyiségünkhöz.

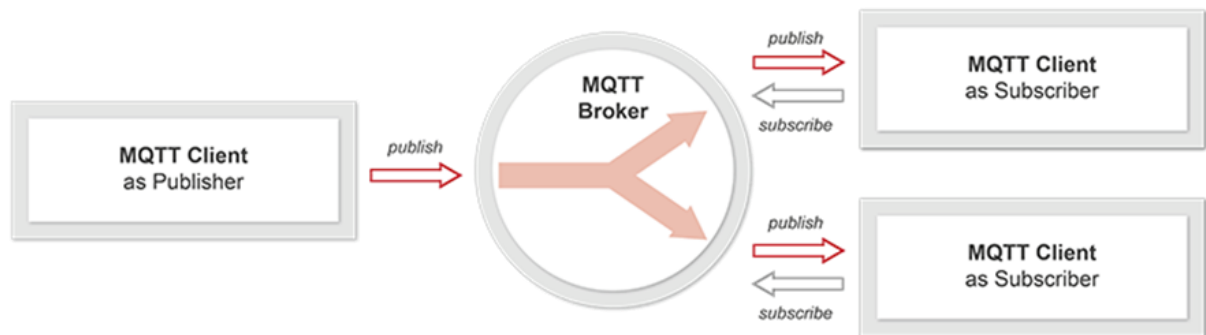
Rendszerünkben egy központi Raspberry Pi eszköznek és három darab ESP8266 eszköznek kell kommunikálnia. Ezt a kommunikációt nagyon jól demonstrálja az M2M kifejezést.

## **8.1. M2M**

Az M2M (Machine to Machine) fő feladata, hogy minimális emberi beavatkozással vagy akár emberi közbeavatkozás nélkül teremtsen kommunikációt a hálózatra kötött eszközök között. Ezen alkalmazási lehetőség az egyik alapja az IoT rendszereknek. Leggyakrabban szenzor adatok átvitelére használják. Az adatokat az autonóm rendszerek feldolgozzák és döntéseket hoznak, majd az M2M segítségével előre definiált automatizált műveleteket indítanak el a beavatkozókon. Protokoll szinten a választásunk az MQTT (Message Queuing Telemetry Transport) protokollra esett, ami egy kis sávzsélességű, nagy késleltetésű protokoll ezért tökéletes az M2M megvalósítására.

## **8.2. MQTT protokoll**

Az MQTT protokoll az ISO/IEC 20922:2016-ban van rögzítve. A szabvány két hálózati szereplőt különböztet meg egymástól. A brókert és a klienseket. A bróker felveszi a klienseket a hálózatra, kezeli azok feliratkozási és leiratkozási kéréseit a topikokról és a publikált üzeneteket továbbítja a topikra feliratkozott klienseknek. Tehát mondhatjuk, hogy egyfajta szerver ahová minden publikált adat beérkezik, majd ő továbbítja azoknak, akik feliratkoztak az adott topikra. A kliens publikálhat adatot topikra, amiben másik kliensek érdekeltek lehetnek, feliratkozhat egy topikra, amiben érdekelt, leiratkozhat topikról és leiratkozhat az egész szerverről is. Tehát ez a rendszer publish/subscribe alapon működik, ahol a kliens lehet egyszerre publisher és subscriber is. A topik lényegében egy téma, ami azonosítja az adatokat.



8.2. ábra MQTT modell [10]

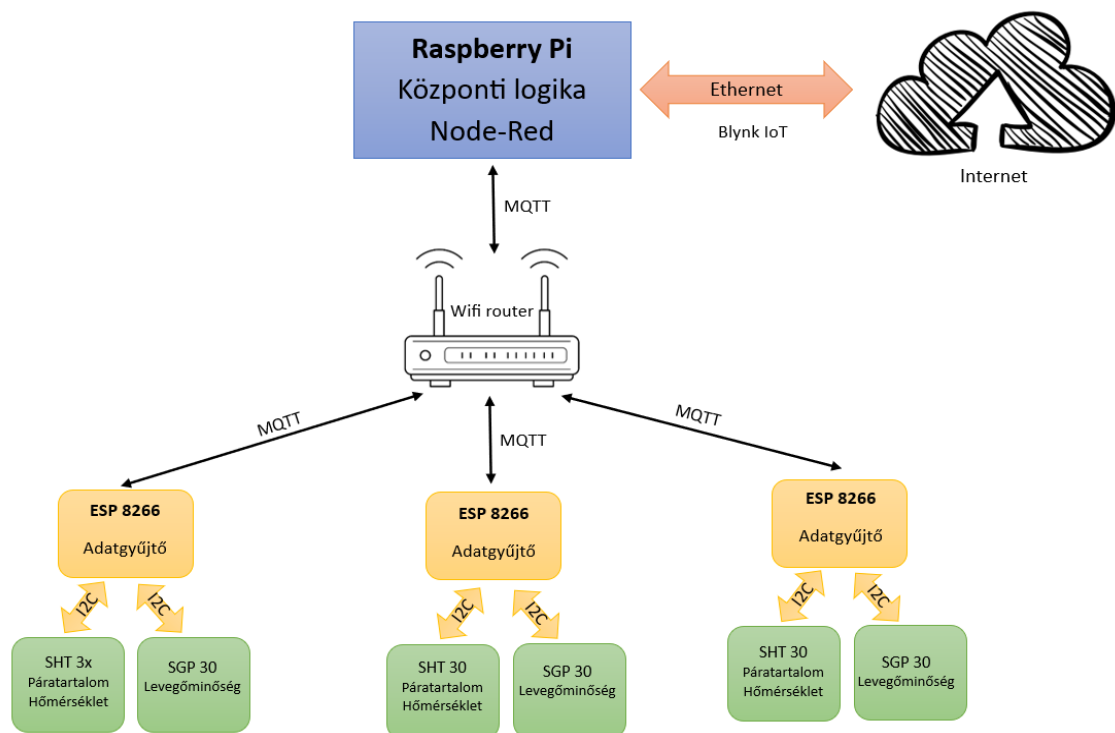
Az adatátvitel általában TCP/IP kapcsolat felett zajlik, ám lehet UDP protokollt is beállítani. A kapcsolat általában titkosítatlan, de itt is van lehetőség változtatásra, TLS titkosítást lehet beállítani. A protokoll három szintjét különbözteti meg a Quality of Service-nek (QoS). Ez azt definiálja, hogy a bróker és a kliens mennyire biztosítja az üzenetek megérkezését. A nulladik szint (QoS 0), amikor semmilyen biztosíték nincs arra, hogy az üzenet megérkezik. Itt a feladó egyszer elküldi az üzenetet, de nem vár semmiféle visszajelzést, hogy a másik fél megkapta. Az első szinten (QoS 1) a küldő elküldi az üzenetet és visszajelzést vár. Ha adott időn belül nem kap visszajelzést, akkor újra küldi az üzenetet. Ez már jobban biztosítja az üzenet megérkezését, ám duplikálódást is előidézhet. Második szinten (QoS 2) a küldő és fogadó közt először egy négy lépéses kézfogás jön létre, ami biztosítja az üzenet átvitelét. Ezután csak egyszer elküldi a csomagot, ami biztosan megérkezik. Minél nagyobb QoS-t állítunk be annál nagyobb lesz az igényelt sávszélesség és a késleltetés is, ezért érdemes megfontolni, hogy mennyire fontos a magas QoS, vagy esetleg elbír egy-két kimaradt üzenetet a rendszerünk. [9]

Lehet használni retain flaget is az üzenetknél. Ha ez be van állítva akkor a bróker miután megkapta az üzenetet és kiküldte a jelenleg a topikra feliratkozó klienseknek, megtartja az üzenetet és ha egy kliens újonnan iratkozik fel a topikra annak is elküldi. Ez hasznos funkció, ha ritkán van csak publikálva adat, mert így sosincs teljesen információ nélkül egyik subscriber sem.

Szintén beállítható a clean session flag is, mely, ha igazba van állítva, akkor a kliens összes feliratkozása elveszik, ha az lecsatlakozik a hálózatról. Ha ez nincs beállítva, akkor feliratkozásai megmaradnak, sőt a nullánál nagyobb QoS -sel rendelkező üzeneteit eltárolja a bróker és kézbesíti, amikor újra csatlakozik.

Will-t azaz végakaratot is lehet beállítani a kliensnek a brókerhez való csatlakozáskor. Ezt az üzenetet fogja elküldeni a bróker amikor a kliens váratlanul lekapcsolódik. Ennek az üzenetnek van topikja QoS szintje és a retain flaget is be lehet állítani, ugyanúgy, mint minden más üzenetnél.

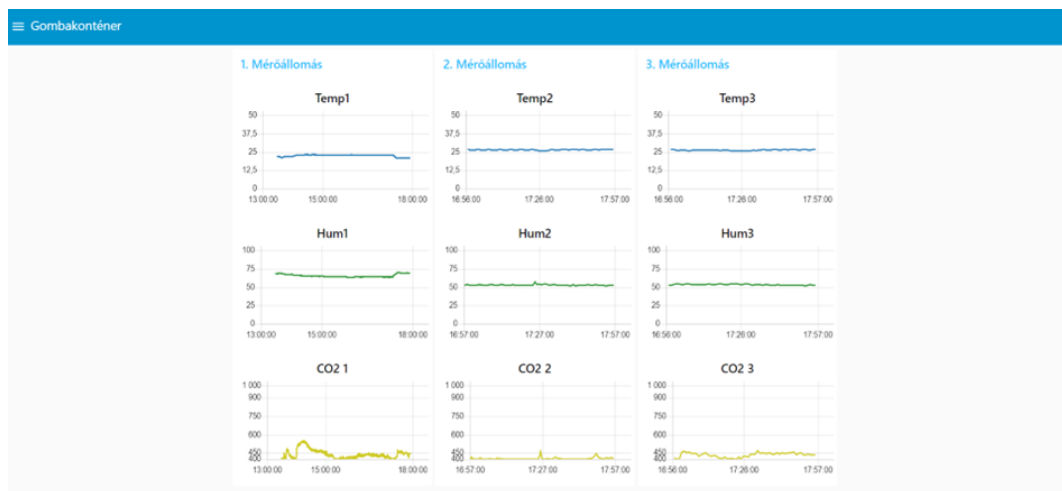
Mi a hálózatunkat úgy építettük fel, hogy a raspberry eszközön fut a bróker. Ezt könnyen meg is tudtuk valósítani, az Eclipse Mosquitto telepítésével. Ez egy ingyenes MQTT bróker szoftver. Mivel a hálózatunk kiterjedése eléggé korlátolt a konténeren belül, így nem volt probléma, hogy azonos alhálózaton legyenek a kliensek, hiszen azonos wifi hálózathoz is csatlakoznak. Kliensek futottak mind az ESP-ken mind a Raspberry-n is. Az ESP kliensek publikálták az adatokat 3-3-3 topikon és ezekre iratkozott fel a Raspberry. Az üzenetek 1 percenként vannak elküldve, ez teljes mértékben elég a pontos nyomon követéshez, ugyanis a szigetelt konténerben nem tudnak olyan nagyon gyorsan változni a tényezők, illetve mind a gombáknak, mind pedig a rendszernek vannak tűréshatárai. Ezeken a topikokon kívül létrehoztunk még a mikrokontrollereket irányító jelzéseknek is topikokat. Erre való üzenetküldéssel távolról lehet újraindítani az ESP-ket. QoS-t csak ezekre a topikokra állítottunk be, mivel törekedtünk a minél gazdaságosabb működésre és egy-egy üzenet kimaradása se okoz nagy problémát a monitorozó rendszerben.



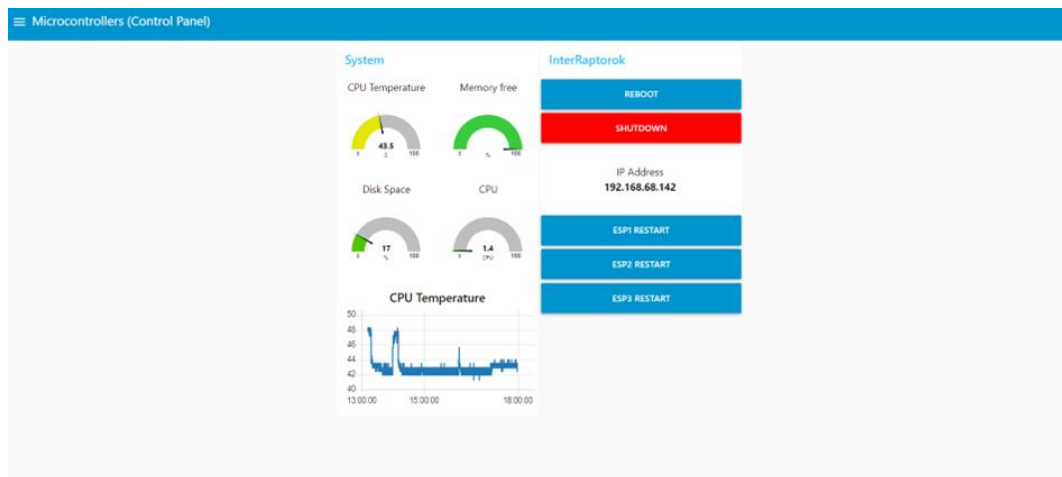
8.3. ábra Mérőpontok megjelenítő felülete



A rendszerünket úgy terveztük, hogy akárhol beüzemelhető legyen, ezért magában foglal egy wifi routert is, ami a vezeték nélküli wifi hálózatot biztosítja. Az eszközeink számára nem szükséges nagy sávszélesség vagy hipergyors adatátvitel ezért bármilyen router megfelel a célra, ami rendeltetés szerűen működik. A rendszer kiépítése közben négy különböző routert is kipróbáltunk, melyből három sajnos nem működött rendesen. Igaz mindegyik használt router volt és egyik se volt fiatalabb 10 évesnél. Negyedszerre már nem routerrel próbálkoztunk, hanem egy mobiltelefon által megosztott Wifi hálózattal. Ez bizonyult a legmegbízhatóbbnak, és ugyan ennél jóval kevesebb beállítási lehetőségünk van, mint a routereknél, alapvetően jól működik. Helyszíni monitorozáshoz és beavatkozáshoz, csak ehhez a wifi hálózathoz kell csatlakoznunk bármilyen eszközünkkel, majd a böngészőbe beírni a „IP:port/ui” kifejezést. (Ahol az IP a Pi IP címe, a port pedig a port, amin a Node-Red fut.) Innen rögtön láthatjuk a három mérőpont mérési értékeit, vagy a mikrokontrollerek aktuális állapotát. Itt újra is lehet indítani őket.



8.4. ábra Mérépontok megjelenítő felülete



**8.5. ábra Mikrokontrollerek visszajelzője**

Amennyiben a távoli monitorozást is használni szeretnénk, ehhez internetkapcsolat szükséges. Ezt a fő egység (Raspberry Pi) Ethernet portjába dugott vezetékes internettel tudjuk biztosítani. Ezen a kapcsolaton publikálja a Pi a Blynk IoT felhőadatbázisa felé az adatokat, melyeket aztán webes nézetből és mobilapplikációból is meg lehet nézni. A mért adatokat időbélyeggel együtt a PI-n csv fájlba is mentjük [8.7. ábra], mert így hosszútávon tárolhatóak az adatok.



**8.5. ábra Webes távoli elérés**



8.6. ábra Mobilapplikációs távoli elérés

1/11/2023	1:40:8	20.76,69.84,420
1/11/2023	1:41:8	20.79,69.94,433
1/11/2023	1:42:8	20.76,69.97,423
1/11/2023	1:43:8	20.79,69.95,429
1/11/2023	1:44:8	20.79,69.93,417
1/11/2023	1:45:8	20.76,69.87,419
1/11/2023	1:46:14	20.77,69.94,424
1/11/2023	1:47:14	20.76,70.04,427
1/11/2023	1:48:15	20.74,69.96,418
1/11/2023	1:49:15	20.76,69.98,408
1/11/2023	1:50:15	20.76,69.98,415
1/11/2023	1:51:15	20.76,69.95,405
1/11/2023	1:52:15	20.76,69.95,403
1/11/2023	1:53:15	20.77,69.96,400
1/11/2023	1:54:15	20.74,69.98,413
1/11/2023	1:55:15	20.73,69.91,403
1/11/2023	1:56:15	20.72,69.87,400
1/11/2023	1:57:15	20.72,69.88,402
1/11/2023	1:58:15	20.76,69.91,400
1/11/2023	1:59:15	20.7,69.88,400
1/11/2023	2:0:15	20.72,69.88,401
1/11/2023	2:1:15	20.72,69.85,400
1/11/2023	2:2:15	20.74,69.91,400
1/11/2023	2:3:15	20.72,69.86,400
1/11/2023	2:4:15	20.73,69.84,400
1/11/2023	2:5:15	20.73,69.82,400
1/11/2023	2:6:15	20.73,69.9,400
1/11/2023	2:7:15	20.7,69.88,400
1/11/2023	2:8:15	20.7,69.86,400
1/11/2023	2:9:15	20.7,69.86,400
1/11/2023	2:10:16	20.7,69.85,400
1/11/2023	2:11:16	20.69,69.88,400
1/11/2023	2:12:16	20.7,69.86,400
1/11/2023	2:13:16	20.69,69.93,400
1/11/2023	2:14:16	20.66,69.84,400
1/11/2023	2:15:16	20.67,69.8,400
1/11/2023	2:16:16	20.69,69.84,400

8.7. ábra Csv fájl tartalma

## 9. Adatgyűjtés a helyszínen

Sajnos a helyszíni adatgyűjtésre még nem volt lehetőségünk, mivel amikor csatlakoztunk, éppen véget ért egy termelési ciklus, majd utána, mielőtt a végleges rendszerrel elkészültünk volna, elszállították egy bemutatóra a konténert. Sajnos erről azóta sincs hírünk, így az új helyszín a kertészeti egyetem gombagarázsa lett, ahol nincsenek jelenleg gombák, de 1-2 héten belül elkezdik a telepítésüket. Gombák híján az elkészült rendszer eddig csak otthoni környezetben volt huzamosabb ideig tesztelve. Több napig állandóan ment, egy napig a lakás különböző pontjaira voltak kihelyezve a mérőegységek, majd később a mérési pontosság megállapítása miatt egy helyre raktuk a három eszközt.

### 9.1. Eredmények

A teszt alatt, amikor külön helyiségben voltak a szenzorok, nem láttunk hibát a mérésekben, szépen követték a nappal-este közben történő hőmérséklet-páratartalom és széndioxid változásokat. Jól látszódott, ha épp rásütött valamelyik szenzorra a nap, vagy bekapcsolt a fűtés. Imitáltuk a konténerben lévő párasítást is. Mivel nincs itthon hidegpárasító berendezésünk, így a mosás utáni száradó ruhákat tettük egy szobába és melléjük tettük az egyik mérőállomást. Látszott a páratartalom folyamatos növekedése. Ezután bekapcsoltuk a szobában a fűtést. Ezzel a teszttel jól tudtuk igazolni a relatív páratartalom hőmérséklet függését. (Minél melegebb a levegő, annál több vízgőzt képes megtartani. Így, ha egy szobában, ahol konstans a levegőben lévő vízgőz, ha elkezdünk fűteni, a relatív páratartalom csökkenni fog.) Azonban amikor egy helyre kerültek az eszközök, akkor kijött egy hiba a rendszerben. A shield-es szerelésű mérőállomások átlagosan 5-6 °C –al többet mértek, mint a nem shield-es változat. Ez azért van, mert ebben az összeszerelésben, az SHT30 szenzor sokkal közelebb van az ESP modulhoz, így az ESP által termelt hő ennyivel befolyásolja a mérést. Ezt úgy tudtuk megoldani, hogy az egyébként shieldes kialakítású szenzort kábelekkel kötöttük az ESP-hez, így távolabb került tőle.

### 9.2. Folytatás

A feladatunk itt még nem ért véget, hiszen a rendszert éles helyzetben még nem volt lehetőségünk kipróbálni. A továbbiakban kivisszük a helyszínre az eszközöket, amikor már elkezdik az újabb gombatermesztést. A gombák telepítését követően bekapcsolják a párasító és légkondicionáló berendezéseket, és azok folyamatosan mennek a termesztés ideje alatt.

Következő lépésként a mérőeszközök által gyűjtött adatok alapján ezen berendezések automatizálását fogjuk megvalósítani. Ezzel jelentős erőforrásokat lehet megtakarítani, ráadásul a gombák is egységesebben, egészségesebben fognak nőni.

## 10. Irodalomjegyzék

1. CSORBAINÉ GÓGÁN ANDREA, DR. PÉK ZOLTÁN: GOMBATERMESZTÉS [ONLINE]. [HTTPS://UJKERT.MKK.SZIE.HU/SITES/DEFAULT/FILES/FILES/GOMBATERMESZT%C3%A9S.PDF](https://ujkert.mkk.szie.hu/sites/default/files/files/gombatermeszt%C3%A9s.pdf). (HOZZÁFÉRÉS DÁTUMA: 2023.10.31.)
2. CSURJA ZSOLT (2018): A MULTIKKAL IS MEGVÍVOTT A MAGYAR GOMBÁS, AKI NAGYBAN JÁTSZOTT ÉS BEJÖTT NEKI [ONLINE]. [https://hvg.hu/gazdasag/20180224\\_A\\_multikkal\\_is\\_megvivott\\_a\\_magyar\\_gombas\\_aki\\_nagyban\\_jatszott\\_es\\_bejott\\_neki](https://hvg.hu/gazdasag/20180224_A_multikkal_is_megvivott_a_magyar_gombas_aki_nagyban_jatszott_es_bejott_neki). (HOZZÁFÉRÉS DÁTUMA: 2023.10.31.)
3. ESP8266 WEMOS D1 MINI: [ONLINE], [HTTPS://MODULSHOP.HU/WEMOS-D1-MINI-V30-ESP8266-IOT-722](https://modulshop.hu/wemos-d1-mini-v30-esp8266-iot-722). (HOZZÁFÉRÉS DÁTUMA: 2023.10.31.)
4. [HTTPS://WWW.NXP.COM/DOCS/EN/USER-GUIDE/UM10204.PDF](https://www.nxp.com/docs/en/user-guide/UM10204.pdf)
5. I2C BUSZ: [ONLINE], [HTTPS://OB121.COM/DOKU.PHP?ID=HU:COMM:BUS\\_I2C](https://ob121.com/doku.php?id=hu:comm:bus_i2c) (HOZZÁFÉRÉS DÁTUMA: 2023.10.31.)
6. KSH: AZ EGY FŐRE JUTÓ ÉVES ÉLELMISZER-FOGYASZTÁS MENNYISÉGE RÉGIÓ ÉS A TELEPÜLÉSEK TÍPUSA SZERINT, [HTTPS://WWW.KSH.HU/STADAT\\_FILES/JOV/HU/JOV0051.HTML](https://www.ksh.hu/stadat_files/jov/hu/jov0051.html) (OLVASVA: 2023.10.02.)
7. KSH: ZÖLDSEGFÉLÉK PIACI TERMELŐI ÁTLAGÁRA [FT/KILOGRAMM], [HTTPS://WWW.KSH.HU/STADAT\\_FILES/ARA/HU/ARA0024.HTML](https://www.ksh.hu/stadat_files/ara/hu/ara0024.html) (OLVASVA: 2023.10.09.)
8. MIKROKONTROLLER ÖSSZEHASONLÍTÁS [ONLINE]. [HTTPS://DIYI0T.COM/TECHNICAL-DATASHEET-MICROCONTROLLER-COMPARISON/](https://diy10t.com/technical-datasheet-microcontroller-comparison/). (HOZZÁFÉRÉS DÁTUMA: 2023.10.31.)
9. MQTT [ONLINE]. [HTTPS://MOSQUITTO.ORG/MAN/MQTT-7.HTML](https://mosquitto.org/man/mqtt-7.html). (HOZZÁFÉRÉS DÁTUMA: 2023.10.31.)
10. MQTT PUBLISH-SUBSCRIBE PATTERN [ONLINE]. [HTTPS://WWW.IPCOMM.DE/PROTOCOL/MQTT/EN/SHEET.HTML](https://www.ipcomm.de/protocol/mqtt/en/sheet.html). (HOZZÁFÉRÉS DÁTUMA: 2023.10.31.)
11. RASPBERRY PI 3 B: [ONLINE], [HTTPS://MALNAPC.HU/YIS/RASPBERRY-PI-3-MODEL-B?SRC=RASPBERRYPI](https://malnappc.hu/yis/raspberry-pi-3-model-b?src=raspberrypi). (HOZZÁFÉRÉS DÁTUMA: 2023.10.31.)
12. SGP30 SZENZOR: [ONLINE], [HTTPS://WWW.MAKERFABS.COM/SGP30-AIR-QUALITY-SENSOR-BREAKOUT.HTML](https://www.makerfabs.com/sgp30-air-quality-sensor-breakout.html). (HOZZÁFÉRÉS DÁTUMA: 2023.10.31.)
13. SHT30 SZENZOR: [ONLINE], [HTTPS://TECHFUN.SK/HU/PRODUKT/WEMOS-MINI-SENZOR-TEPLOTY-A-VLHKOSTI-I2C-SHT30-SHIELD/?LANG=HU&CURRENCY=HUF&GAD\\_SOURCE=1&GCLID=Cj0KCQjWY4KQBHD0ARIsAEbCT6GcKeiGUYG7LJo-](https://techfun.sk/hu/produkt/wemos-mini-senzor-teploty-a-vlhkosti-i2c-sht30-shield/?lang=hu&currency=huf&gad_source=1&gclid=Cj0KCQjWY4KQBHD0ARIsAEbCT6GcKeiGUYG7LJo-)

LH\_RMRX70Tp0GN0NZIRNqX0JRSCYcKFBy49B8UGAALUTEALw\_wCB.

(HOZZÁFÉRÉS DÁTUMA: 2023.10.31.)

14. SHT3X SZENZOR: [ONLINE], [HTTPS://WWW.FRUGO.HU/SHT31-HOMERSEKLET-ES-PARATARTALOM-ERZEKELO-KITORESITABLA-IDOJARASI-MODUL-SHT31-SHT31-D/P-215553397-459699652?LANGUAGE=HU&AC=KELKOOCSS&ASC=PMAX&GAD\\_SOURCE=1&GCLID=CJ0KCQJwY4KqBHD0ARIsAEbCt6hDEH-DHTE\\_2R\\_OiZTnQHEJ\\_kJYVVQVg-vLKL\\_n78MHNYQ6NjVWzWGAaHQGEALw\\_wCB](https://www.fruugo.hu/sht31-homerseklet-es-paratartalom-erzekelo-kitoresitabla-idojarasi-modul-sht31-sht31-d/p-215553397-459699652?language=hu&ac=kelkooCSS&asc=pmax&gad_source=1&gclid=CJ0KCQJwY4KqBHD0ARIsAEbCt6hDEH-DHTE_2R_OiZTnQHEJ_kJYVVQVg-vLKL_n78MHNYQ6NjVWzWGAaHQGEALw_wCB). (HOZZÁFÉRÉS DÁTUMA: 2023.10.31.)
15. TÖRÖK P. (2019). TITKOS ÜZENET SZÁLL A SZÉLLEL! (IOT-BEN HASZNÁLT VEZETÉK NÉLKÜLI ADATÁTVITELI TECHNOLÓGIÁK ÖSSZEHASZNÁLÁSA), HADMÉRNÖK, 14. ÉVFOLYAM, 3. SZÁM, P.130-146.