

Csőtörés-jelző rendszer

Pipeline

Székely Marcell

Felkészítő tanár: Horváth-Varga Péter

Kempelen Farkas Gimnázium, 1223 Budapest Közgazdász u. 9-11.

1. Bevezetés

Ma minden lakóházban, irodában, gyárban és egyéb intézményben jellemzően van kiépített vízvezetékrendszer. Egy váratlan csőtörés esetén akár komoly károk is keletkezhetnek az épületben, az elektromos berendezésekben vagy a berendezési tárgyakban. Rosszabb esetben egy épület hosszabb időre is lakhatatlanná válhat vagy egy irodát csak korlátozottan tudnak használni a helyreállítások ideje alatt. Jól látható tehát, hogy egy csőtörés milyen mértékű károkat okozhat – én ezen károk mérséklésére vagy kiküszöbölésére szeretnék megoldást nyújtani.

A probléma megoldására egy olyan eszközt terveztem, ami időben jelezni tudja a váratlanul megnövekedett vízfogyasztást, így ezzel minimálisra csökkentve egy esetleges csőtörés okozta károkat. Az egyik célnak azt tűztem ki, hogy ezt a rendszert bárki egyszerűen, szakember segítsége nélkül is tudja telepíteni, ehhez ne kelljen megbontania a vízvezetékrendszert. A másik célom pedig az volt, hogy egy csőtörést minden esetben jelezzon, a téves riasztások minimalizálása mellett. Ezeken túlmenően egy olyan internetes felület elkészítését is terveztem, melyen keresztül a felhasználó könnyen, bárhol is elérheti az eszközt, hogy távolról is elvégezhesen rajta beállításokat, vagy megtekinthesse az eszköz állapotát.

Ahogy írtam, egy egyszerűen, a vízrendszerre megbontás nélkül beszerelhető eszközt terveztem. Ehhez az kellett, hogy kívülről meg lehessen állapítani, hogy egy csőben folyik-e a víz vagy sem. Erre a célra léteznek ultrahangos áramlásmérők, viszont ezek ára igen magas, emiatt egy másik mérési módszert kellett alkalmaznom. Egy vízvezetékre erősített mikrofonnal a mért frekvenciákból nagy biztonsággal meg lehet határozni, hogy éppen áramlik-e víz, sőt még akár azt is, hogy milyen intenzíven. Azzal a további céllal is készítettem ezt a prototípust, hogy a későbbiekben felhasználható legyen más problémák megoldására is (pl. jellemző hangfrekvenciát kibocsátó eszköz vizsgálatára és hiba-előrejelzésére).

2. Probléma megoldásának menete

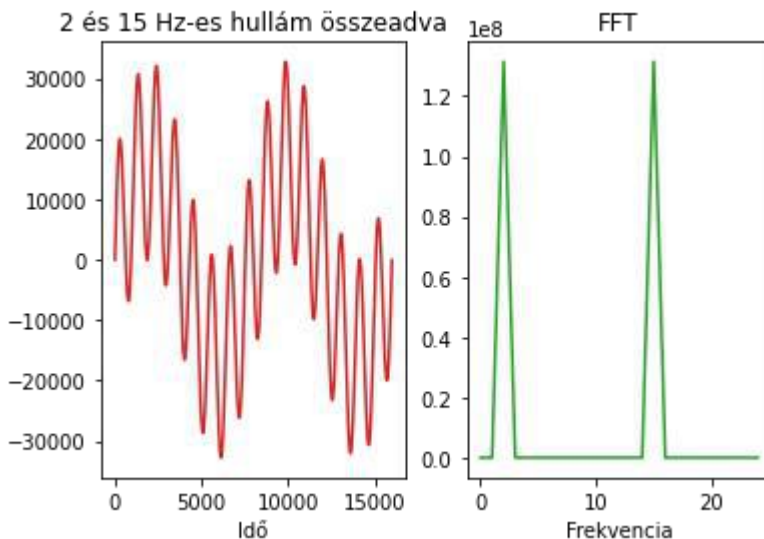
A megoldás első állomása a felhasznált eszközök kiválasztása volt. A szenzor adatainak gyűjtésére és feldolgozására egy STM32 Nucleo-144-es panelt választottam, melyen egy 32 bites ARM Cortex M4-es MCU működik, amit C-ben programoztam. A 32 bites mikrokontrollerre azért volt szükségem, mert a

nagy mennyiségű hanganyag feldolgozásához és tárolásához nagy számítási és tárolási kapacitás szükséges. A hang felvételére pedig egy Adafruit digitális mikrofont alkalmaztam. Ezen kívül szükségem volt még egy wifi kiegészítőre, mert így nem kell Ethernet kábelt kivezetékelni az eszközhöz, így még egyszerűbben felszerelhetővé válik. Az eszközök összeszerelése után kezdődhetett el a programozás.

A kihelyezett eszközön felül még fejlesztenem kellett egy olyan szolgáltatást is, melyhez az eszköz képes kapcsolódni és a feldolgozott adatokat felküldeni. Ezen felül ennek a szolgáltatásnak a feladata a beérkező adatok alapján döntést hozni és riasztási eseményeket kiváltani. Szintén ez a komponens biztosítja a webes konfigurációs felületet is. Ennek a szolgáltatásnak a futtatására Amazon felhőszolgáltatást választottam (AWS), mivel ezzel tudtam a leggyorsabban megoldani a feladatot.

2.1. Adatok fogadása, feldolgozása a mikrokontrolleren

Első lépésnek a mikrofonról I2S protokollon keresztül be kellett olvasnom az adatokat. Ezek után következett a hang frekvenciájának kiszámolása, hiszen a frekvenciából tudunk következtetni a folyadék áramlására. A mikrofonból a hang az idő függvényében érkezik, ebből viszont még nem lehet tudni a hang frekvenciáit. Erre kellett alkalmaznom az FFT algoritmust. Ez egy bonyolult matematikai művelet, amivel kiszámolhatjuk a bejövő hang frekvenciáit.



1. ábra: FFT és szinusz hullám

Az 1. ábrát Pythonban példának generáltam 2 Hz-es és 15 Hz-es szinusz hullám összegéből, erre pedig elvégeztem az FFT-t. A jobb oldali

grafikon két kiugrása 2 Hz-nél és 15 Hz-nél van, így mutatja meg a két frekvenciát, amiből az első hullám áll.

Az FFT alkalmazását először nem a valós hanganyagon teszteltem, hanem generáltam szinusz hullámokat, melyekre elvégeztem az FFT-t. Amikor a generált hullámnak megfelelő frekvenciát kaptam eredményül bebizonyosodott az algoritmus helyessége.

2.2. Az adatok kiértékelése és tárolása

A FFT számítás után a frekvencia adatokat ki kell értékelni. Minden vízvezetéknek más hangja van, így először muszáj egy kalibrációt végezni az eszközön. A folyamatot a webes felületen lehet elindítani. A kalibráció után a rendszer megjegyzi, hogy melyik frekvenciatartományban kell figyelni a kiugrást, így mindig tudni fogja, hogy mikor folyik a víz az adott vezetéken. Az idősoros vízáramlási adatokat (hogy mikor folyik vagy nem folyik a víz) Amazon EC2-ben futó InfluxDB adatbázisban tárolom. Az eszköz wifi kapcsolaton keresztül jelentkezik fel a felhőbe és rendszeresen küldi a telemetriai adatokat. Ezeket az adatokat folyamatosan figyelni kell, hogy időben értesíteni tudjuk a felhasználót egy csőtörésről. Erre a célra egy Python-ban írt Lambda függvényt hoztam létre, mely aktiválását egy ütemező váltja ki.

2.3. Riasztási események kezelése

A beérkező adatokból, a beállított paraméterek alapján az alkalmazás algoritmikus megoldással döntést hoz, hogy szükség van-e riasztási esemény kiváltására. Amennyiben a felhasználót értesíteni kell, az alkalmazás ezt emailen és SMS üzeneten keresztül hajtja végre. Email és SMS küldésére külső szolgáltatásokat vettem igénybe (AWS SES, Seeme SMS)

2.4. Webes felület fejlesztése

Egy Amazon CloudFront által kiszolgált webes felületet is kellett fejlesztenem, hogy a felhasználók könnyen hozzáférjenek az eszköz beállításaihoz. A felület hátterét egy Pythonban írt Rest API adja, ami szintén Lamdaként fut Amazonon belül. A felhasználó adatait és beállításait Amazon DynamoDB-ben tárolom, ahonnan ezeket a kitelepített eszköz leszinkronizálhatja magának.

A felületről lehet a hang konfigurációt elindítani, beállítani az értesítendő emailt és telefonszámot, ezen kívül a riasztásokat is testre lehet szabni.

3. Elért eredmények

A kitűzött célok szerint sikerült elérnem, hogy az eszköz minden esetben érzékeltetnie tudja, ha egy vezetéken víz folyik keresztül. Az internetes felületen be lehet állítani riasztásokat, például időkorlátot külön napszakokra vagy akár konkrét napra is, amikor, ha a beállított paraméternél hosszabb ideig folyik a víz akkor értesíti a rendszer a felhasználót. Emellett az értesítés módját is meg

lehet adni, hogy emailben vagy SMS-ben is jelezzen. A rendszer így nagy megbízhatósággal tudja a felhasználót értesíteni egy adott vízvezeték hibájáról. Ezzel a módszerrel minden nagyobb kárt ki lehet küszöbölni, ahol egy adott vízvezetékben nem folyamatosan áramlik a víz.

Az eszköz arra még nem alkalmas, hogy egy irodaház fő vízvezetékére felszereljük, hiszen nagy valószínűséggel ott mindig folyamatos folyadékáramlás a sok felhasználó miatt, így ott egy algoritmikus rendszer nem tudna jól működni. Időtúllépésnél valószínűleg mindig téves riasztások érkeznének az egyfolytában áramló víz miatt, bár egy éjszakai csőtörés érzékelésére a rendszer itt is alkalmas lenne, mivel akkor már nem tartózkodnak sokan az épületben, így nem lenne állandó vízfogyasztás. Jövőbeli tervem, hogy az algoritmikus módszert gépi tanulásra cseréljem, így az eszköz megtanulná, hogy adott időben egy vezetéken mennyire folyik a víz. Ebből anomáliákat keresve következtetne, hogy a szokásostól eltérő a vízhasználat, ilyen esetben pedig riasztást küldene.

Jelenleg az eszköz fő felhasználási területei a lakóházak vízvezetékrendszere, vagy nagyobb létesítmények különálló kisebb részei, például egy konyha vagy mosdó. Ezeken a területeken a bemutatott rendszer sikeresen alkalmazható a jelentős anyagi károkat okozó lehetséges csőtörések jelzésére.