

„Raspberry Kreatív: Te és a Raspberry-mit tudtok kihozni egymásból?”

Pellettüzelésű kazánt vezérlő rendszer

Kovács Zoltán
KOZWABG.KEFO

2016.06.16.

KF-GAMF

Feladat:

A cél az volt, hogy valamit kezdjünk a Raspberry-vel. Én szerencsére mertem nagyot álmodni, ugyanis egy már meglévő, otthoni központifűtés tervezett továbbfejlesztésére használtam föl a kiadott panelt. A meglévő fatüzelésű rendszer is rendelkezik MCU-s vezérléssel, amit még 2012ben kezdtem fejleszteni. A jelenlegi a hőmérsékletek naplózásán kívül a használatimelegvíz (HMV) előállítást szabályozza.

Az elképzelés:

-Pellet, gabona hulladék elégetéséhez kell egy égőfej, amiben levegő befúvás segítségével ég el a tüzelőanyag, amit a fűtés igénynek megfelelően kell adagolni.-Hőmérőszenzorok és kapcsolórelék hadát kell az RPI-nek kezelnie. Az előbbivel –DS18B20- már régebben megismerkedtem, a korábbi rendszer gyártása során. Ennek nagy előnye, hogy digitális és csupán csak egy érpár kell a kommunikációhoz. Külön tápfeszültséget sem igényel, az adatvezetéken megy a táp is, amikor kell neki.

A bemenetek

Hőmérőszenzorok:

- Kazán vízhőfok (DS18B20)
- Kazán füstgáz hőmérséklet (K-típusú)
- Hőcserélős villanybojler vízhőfok (DS18B20)
- Külső hőmérséklet (DS18B20)
- Belső hőmérséklet (DS18B20)

Egyéb:

- A felhasználó által beállítható paraméterek

A kimenetek

Relék:

- Tüzelőanyagbehordó motor
- Ventilátor motor
- Gyújtó berendezés
- Hamukihordó motor

- HMV szelep

Egyéb:

- A mért adatok megjelenítése

Megvalósítás:

Kezdetben Android mozgatta a fantáziámat a grafikus felülete miatt, de mint kiderül, az eszköz nem támogatja. Egyéb alternatívák voltak a LINUX és a Windows IoT. Utóbbit hamar elvetettem a szoros Win 10-hez való kötődése miatt. Korábban PIC Basic Pro-ban (PBP) majd Arduino-ban programoztam ezt azt, ezért is esett a választás a Qt Creator-ra. Van benne grafikus felhasználófelület (GUI) tervező és nem utolsó sorban rengeteg oktatóvideó az interneten.

Így LINUX lett az alapja, amivel meg is kellett küzdenem. Ez előtt sosem kellett „pingvinel” foglalkoznom, de most előtt az idő.

Miután két hete próbálkoztam a Cross Compile funkció összehozásában Windows,LINUX-RASPBIAN között. Ennek oka az volt, hogy VNC-n keresztül a program használhatatlan billentyűzet kiosztást alkalmazott. Végül sikerült találnom egy X11VNC nevű programot, ami az RPI 0 képernyőjét osztja meg, normális billentyűzettel.

Idő közben vásároltam egy 7"-os LCD kijelzőt (800x480) amit HDMI-vel lehet összekötni a miniszámítógéppel. LINUX alatt ezzel is küzdeni kellett a normális megjelenítés miatt.

A Qt alapból nem kezeli a GPIO-kat, ezért kellett alkalmazni egy WiringPI nevű könyvtárat. Ez még az I2C és SPI kommunikációs protollokat is kezeli, sajnos a DS szenzorokhoz szükséges One Wire-t nem. A PBP alapból tudta ezt, de hosszabb vezetéknél már nem várt eleget a válaszra, ezért már akkor is lekellett modellezni a kommunikációt saját „függvénnyel”. Most sem lett másképp az akkori programom alapján sikerült a OW-t lekezelni a Qt-ban. Ezzel is voltak persze problémák, hogy csak egyet említsek: A Qt Sleep-je nem képes pontos időzítés ként szolgálni, így erre is rá kellett jönnöm, hogy 10us az neki néha 120ms. Ez már bőven elég ahhoz, hogy haza vágja a szenzorral való beszélgetést. Ezt sikerült egy Timer-rel megoldani, így írni-

olvasni már tudtam. A szenzorok mivel digitálisak ezért címezhetők, ezért lehet egy érpáron az összeset külön-külön kezelni. A címek úgy nevezett ROM kódok amik tartalmazznak egy 48 Bites egyedi sorozatszámot. Ezeket legegyszerűbben egyesével tudjuk kiolvasni a szenzorokból, de van egy elegánsabb módja is, amikor egy algoritmus segítségével a program az összes egy vezetéken lévő szenzort be tudja azonosítani. Ezt is sikerült megvalósítani egy kis fejtörés után.

A ROM-ok kiolvasása után a program egy txt-fájlba menti az olvasottakat egy-egy megadott névvel, ahonnan a többi indulásnál betölti ezeket. A hőmérséklet kérésre érkezik 2 byte-on amit egy kis bitműveletekkel akár 12 bites felbontásban kapunk meg(-55°C - +125°C). A 2 byte mellett érkezik még hét, köztük egy a CRC kód az adatvesztés ellenőrzéséhez.

Érintő panel

Eleinte megfordult a fejemben, hogy a kijelző mellett majd gombokat helyezek el a kezeléshez, de amint világossá vált számomra, hogy néhány ezer forintért lehet kapni megfelelő méretű touch panelt, így elvetettem a mechanikus technikát. Az érintő kontrollere USB porton keresztül kapcsolódik a málna PC-re. Egy kis LINUX-os telepítgetésen és kalibráláson kívül nem is kell más a működéséhez.

GUI

Az volt az elképzelés, hogy majd rajzolok egy szépet mondjuk Photoshopban. Rá kellett jöjjek, hogy nem fogok, mivel nem tudok. Ezután nézegettem weblap sablonokat, -közben kipróbáltam a Qt-ban lehet képet beilleszteni és fölé pl egy átlátszó gombot rakni- gondoltam majd összeollózok valamit. Amiket ingyen találtam azok kis felbontásúak voltak és nem feleltek meg az elvárásnak. Az idő teltével, láttam oktató videókat Qt designer használatáról és rájöttem, hogy nem kell össze ollóznom, meg tudom szerkeszteni Qt-ban. Már csak a színvilág volt a kérdés, ehhez segített az egyik netes template. Végül annyira

eltértem az eredetitől, hogy ma már szinte csak a háttér szín van át emelve, a többi színt én választottam.

Csomagolás:

Szerencsére a nagyobbik 240x190x90-es dobozt vásároltam hozzá, ebben éppen elférnek az alkatrészek. Fúrást, vágást, reszelést követően lefestettem a dobozt. Száradást követően a kijelzőt valahogyan rögzítenem kellett a doboz tetején, ehhez semmi felfogatási pont nem volt a dobozon. Végül a kijelző panel teljes szétszerelését követően forrasztva lettek a rögzítőkonzolok az LCD hátsó kasztnijához, ami a nagyon gyors oxidáció miatt jóval nehezebben ment, mint ahogy hangzik.

Panel:

Eagle áramkör tervezővel (és laminálóval) készítettem egy panelt amire a perifériák tudnak csatlakozni és az RPI egy 40eres IDE vezetékekkel kapcsolódik rá. A nyákon helyet kapott egy 7805 feszültség stabilizátor, ami mint kiderült még rezsóként is funkcionált rajta. Nem jutott el a tudattomig, hogy ez nem egy néhány mA-t fogyasztó PIC. Ezt követően beépítésre került egy 3A-es Step-Down konverter panel, és egy közvetett levegőhűtés a dobozba.

Szimulátor:

A programíráshoz és a demonstrációhoz szükségessé vált egy kazán szimulátor, még át nem alakított kazán híján. Ebben egy Arduino figyeli a relékimeneteket és PWM-el vezérel egy izzót, ami a láng melegét szimulálja. Az izzó közvetlen közelében van a füstgáz szenzor, majd ettől távolodva a kazán és a bojler szenzorra.

Program:

Sikerült megoldanom a többszálon való futtatást, jelenleg egyszerre három szálon fut a program. Nem aktív még minden funkciója és bugos is itt-ott, de csak idő kérdése és működni fog minden.

Jelenlegi funkciók:

- Idő-dátum kijelzése
- 5 szenzor mérése és kijelzése
- Begyűjtás és fűtés kezdetleges vezérlése
- Menüben kereshető szenzorok.

Tervezett funkciók:

- Tüzelőanyag szintszámítás a napi tartályban
- Paraméterek állítása menüben
- Hűtő ventillátor vezérlése
- Webes, okostelefonos elérés nem csak vnc-n keresztül.
- Értékek loggolása külső szerverre.