Műszaki Informatika Verseny Középiskolásoknak 2014

Szegedi Tudományegyetem TTIK Műszaki Informatika Tanszék

EZT RAKTUK ÖSSZE!

Szegedi Tudományegyetem
2014

ÚJ SZÉCHENYI TERV

Bíráló bizottság:

Dr. Gingl Zoltán SZTE TTIK ITCS Műszaki Informatika Tanszék egyetemi docens

Dr. Pletl Szilveszter SZTE TTIK ITCS Műszaki Informatika Tanszék főiskolai tanár

Dr. Mingesz Róbert SZTE TTIK ITCS Műszaki Informatika Tanszék egyetemi adjunktus

Engedi Antal SELECTRIC Mérnöki Iroda Kft. cégvezető, Mérnöki Kamara, Kereskedelmi és Iparkamara tag

Szervező:

Dr. Kincses Zoltán SZTE TTIK ITCS Műszaki Informatika Tanszék egyetemi adjunktus

Támogatóink:

National Instruments, hungary.ni.com

Evosoft, www.evosoft.hu/

OMRON, omron.hu/hu/home

Evopro, www.evopro.hu/

Magyar Mérnöki Kamara, mmk.hu/

Szent-Györgyi Albert AGORA, www.agoraszeged.hu

A kiadvány a TÁMOP-4.2.3-12-1/1/KONV-2012-0035, Tudományos eredmények elismerése és disszeminációja a Szegedi Tudományegyetemen pályázat keretében készült.

Előszó

A Szegedi Tudományegyetem Műszaki Informatika Tanszéke a TÁMOP-4.2.3-12/1/KONV-2012-0035 Tudományos eredmények elismerése és disszeminációja a Szegedi Tudományegyetemen című pályázat keretein belül megrendezte a I. Műszaki Informatikai Versenyt középiskolás diákok számára.

A célunk egy olyan verseny megszervezése volt a műszaki informatika témakörében, melyen a tehetséges, kreatív diákok lehetőséget kaphattak eredményeik bemutatására. A verseny során a diákok közvetlen kapcsolatba kerülhettek a Szegedi Tudományegyetemmel és az ott dolgozó oktatókkal. A verseny hosszabb távon a hallgatói tudásszint javításához is hozzájárulhat, támogatja a tehetséggondozást, valamint a műszaki informatikai képzés népszerűsítését.

A verseny keretein belül nem kötöttük meg a diákok kezét és kreativitását, bármilyen probléma, feladat megoldásával kapcsolatos pályamunkával nevezhettek. Egyetlen kikötés volt csak: a feladat ne csupán programozási feladatból álljon. A diákoknak meg kellett építeni, vagy már meglévő elemekből össze kellett állítania egy rendszert (elektronikai elemek, motorok, fejlesztő kitek, érzékelők, stb...), amely működtetéséhez szükséges szoftvert is a diákoknak kellet elkészítenie.

A versenyre összesen 23 darab igen színvonalas pályamunka érkezett, melyből a verseny bírálóbizottsága kiválasztotta a 10 legjobbat, melyek bekerültek az Agora Informatóriumában 2014. február 28-án megrendezésre került döntőbe. Ez a kis kiadvány a 10 legjobb pályamunkát foglalja egy kötetbe.

Szeged, 2014.02.18.

Dr. Kincses Zoltán
Kiadvány szerkesztője

Tartalom

Fakla Dávid, Majoros Attila: Szaktudással kinyílik a világ… 5

Kiss Tamás, Csipak Levente: eMotion, avagy a GPS tehénpásztor 7

Papp Zoltán, Tusa Balázs, Tóth-Albert Tibor: Háromdimenziós led mátrix 9

Hegyesi Donát Sándor: Pókerező Robot 11

Vincze Zoltán Dániel: Légbőlkapott 13

Bálint Karola, Gilicze Kristóf, Sebők Attila: A XXI. század menekítő csapata 15

Kiss Máté: Róka-nyúl játékgép 17

Kiss Ádám, Szabó Benedek: Mérések mikrokontrollerrel napkollektor optimalizáláshoz 19

Bosznai Tamás, Remete Bence: Reboss csapat 21

Alekszejenkó Levente, Ugrin Dániel: A Tóth Árpád Gimnázium új csengője, a tAGBell 23

Program

9:00	**Regisztráció**
10:00	**Délelőtti szekció**
10:00	*Legorockers: „A XXI. század menekítő csapata”*
10:20	*Materex: „Róka-nyúl játékgép”*
10:40	*Murphy's Hands: „Mérések mikrokontrollerrel napkollektor optimalizáláshoz”*
11:00	*Reboss csapat: „Reboss csapat”*
11:20	*BolyaiSokk: „eMotion, avagy a GPS tehénpásztor”*
11:40	*ESZI szakkör: „Háromdimenziós led mátrix”*
12:00	**Ebéd**
13:00	**Délutáni szekció**
13:00	*Hegyesi Donát: ”Pókerező robot”*
13:20	*TeamTAG: „A Tóth Árpád Gimnázium új csengője, a tAGBell”*
13:40	*Légbőlkapott: „Légbőlkapott”*
14:00	*Álomépítők: „Szaktudással kinyílik a világ…”*
14:00	**Látogatás az Agóra Informatika Történeti Kiállításra**
15:00	**Eredményhirdetés**
15:30	**Igény szerint látogatás az SZTE Műszaki Informatika Tanszéken**

Szaktudással kinyílik a világ...

Álomépítők

Fakla Dávid, Majoros Attila
Kandó Kálmán Szakközépiskola, Miskolc, Palóczy u. 3.

1. Bevezetés

Műszaki informatikus tanulóként arra gondoltunk, hogy az eddig megszerzett ismereteink alapján, összeállítunk egy vezeték nélkül vezérelhető eszközt, melyre a legalkalmasabbnak egy távirányítós autót találtunk. Rengeteg RC-s autót lehet kapni a boltokban, de olyan, ami sokat tud, nagy a teljesítménye és elérhető áron van, olyan kevés van. Így mi megálmodtunk egy olyan prototípust, amely - követve jelenkor fejlődő technológiáját - akár számítógépről vagy mobiltelefonról is működtethető. Fellelkesülve az ötleten máris megfogalmazódott bennünk, hogy olyan járművet építünk, ami könnyen irányítható, alacsony költségvetésű, magas teljesítménnyel bír és okos.

2. A probléma megoldásának menete

A távirányítós autónk elektronikájának elkészítéséhez sorra átvizsgáltuk a lehetséges kontrollerek típusát és felépítését. Azt kerestük, hogy melyik fejlesztői környezet az, amelyiket megengedhetjük magunknak - saját pénzből - és kielégíti az igényeinket. Csapatunk végül, az Atmel processzor köré épülő Arduino eszközt választotta, mert az előzetes ismereteink meg voltak hozzá és otthoni munkálatokhoz is kiválóan alkalmas. Név szerint egy Ardiuno UNO (1. ábra) fejlesztő eszközt használunk, amiben jelenleg egy Atmega328P mikrovezérlő található. Az eszköz alkalmazásának a megtanulása volt a legnagyobb kihívás számunkra, hiszen úgy kerek egész a dolog, ha a tudásunkat párosítjuk a gyakorlathoz, és az eszköz működését a valóságban is megtapasztaljuk. Sajnos a Wifi modul használatát nem engedhettük meg magunknak, így maradt a bluetooth (2. ábra), vagyis virtuális COM porton való kommunikálás az autóval.

Először a hardvert szereztük be, hogy legyen mit vezérelni. Maga az autó működtetése igen egyszerű, két darab H-híddal van megoldva. Az autó meghajtását illetően, apránként haladtunk előre, először szoftveresen, direkt módon hajtottuk a motort két különböző irányba. Az Arduino 5V tápfeszültség mellett csak maximum 40 mA áramot képes kiadni, ami kevésnek bizonyult, így meg kellett oldanunk azt, hogy legalább a hátsó motor erős meghajtás kapjon.

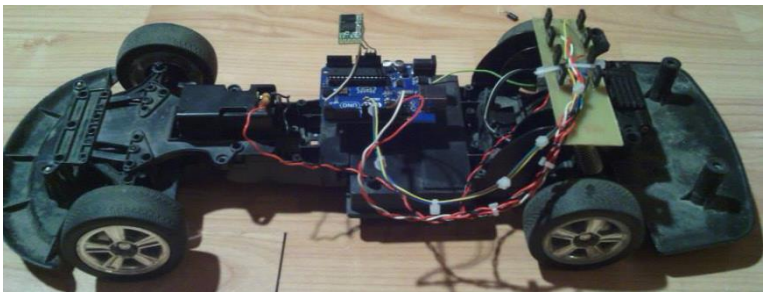
1. ábra. Az Arduino Uno

2. ábra. A bluetooth modul

3. ábra. A motor shield

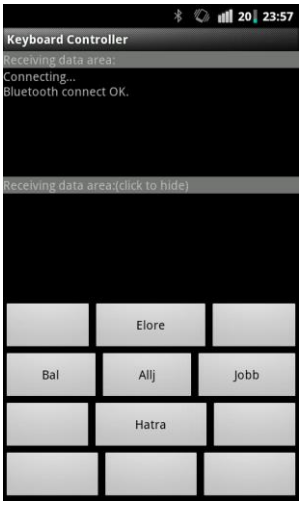
Ennek a problémának az ismeretében elkezdtünk saját hardvert készíteni. A NYÁK-ok (nyomtatott áramkörök) tervezése, gyártása és építése során rengeteg verzió készült az autó elektronikájából. Folyamatos tesztelések zajlottak, olykor új programot is kellett írnunk, mire sikerült kifejleszteni a megfelelő prototípust. BC139 és 140-es komplementer tranzisztorpárral valósítottuk meg a vezérlést, viszont a nagy motorokat ez se tudta kellőképpen meghajtani, de itt már lehetett a H-hidat az Arduinoval irányítani. Így szépen lassan kialakult bennünk, hogy hogyan is lesz megvalósítva a kész autó. A H-híd teszi lehetővé nekünk, hogy két irányba tudjuk vezérelni a motort. Végül is kész lett a két híd és összeálltak a teszt modellek (4. ábra) Ezzel a megoldással már viszonylag jól működött az elektronika, de a hátsó motor nagy indulási áram felvételét nem bírták huzamosabb ideig a kis tranzisztorok, így más megoldás kellett, jelen esetben nem erősebb tranzisztor vagy MOSFET, hanem egy sokkal esztétikusabb megoldás, egy motor shield (3. ábra).

Kicsi, kompakt, hűtés nélkül 15A-ig simán terhelhető és pont elég az autóhoz. Lámpákkal is felszereltük a járművet, ami extra kiegészítőnek számít.



4. ábra. Az első teszt modellek

A távvezérlést mind számítógépről, mind pedig okostelefonról meg lehet valósítani (5. ábra), azonban az utóbbi hordozhatóság szempontjából mindenféleképpen kényelmesebb, mert a kezünkben is elfér. Jelenleg Android és Windows operációs rendszerek alatt irányítható az autó. A hatótávolság - a bluetooth miatt - körülbelül 10-15 méter. Az autóban lévő program folyamatosan olvas a soros porton, és ha egy olyan karaktert küldünk, ami mögé egy függvény van írva, akkor aszerint cselekszik, például ha egy 'E' betűt küldünk, akkor előre megy, és amíg 'E' a kapott érték, addig az autó folyamatosan előre halad.



5. ábra. A távirányító

3. Elért eredmények

Viszonylag ismeretlen dolog volt számunkra a mikrovezérlők gyakorlati alkalmazása, de az, hogy a tanítási órák után estig ott maradtunk az iskolában, meghozta a gyümölcsét. Hamar belerázódtunk, és még ha kis lépesekben is haladva, de elindultunk a cél felé. Természetesen a problémákat igyekeztünk minél gyorsabban kiküszöbölni, megoldani, és amikor már megértettük, hogy mi miért van, hogyan működik, akkor már könnyebben vettük az akadályokat. Az Arduino eszköz által szerzett tapasztalatok alapján rájöttünk, hogy rengeteg bővítési és alkalmazási lehetőség van ebben a fejlesztő környezetben, és nagyszerű dolog lenne ezt kihasználni.

Az ötletünket a háztartásokban gondoltuk felhasználni, mivel sok mindent lehet már egy lakásban vezérelni és szabályozni elektronikával. A fűtési rendszert lehet ilyen Arduino eszközzel működtetni, ahol a programot mi magunk írjuk, és úgy állítjuk be a paramétereket, ahogy a család számára a legkényelmesebb.

Az Arduino eszköz bővíthetőségére számos lehetőségünk van, fel lehet szerelni rá érzékelőket, jeladókat, kijelzőket, amelyeket a későbbiekben mi is szeretnénk kipróbálni, illetve meg is valósítani.

Rengeteget tanultunk a projekt készítése alatt, melyeket későbbi tanulmányaink során fel fogunk használni, illetve már a gyakorlatban is könnyebben elboldogulunk hasonló feladatok megoldásakor.

eMotion, avagy a GPS tehénpásztor

BolyaiSokk

Kiss Tamás, Csipak Levente
Bolyai Tehetséggondozó Gimnázium és Kollégium

1. Bevezetés

Munkánk célja az, hogy valamilyen módon könnyebbé, korszerűbbé tegyük a különböző házi- illetve haszonállatok nyomon követését, őrzését. Az elkóborolt állatok komoly fejtörést tudnak okozni a gazdának, ezért a fejlesztés fő célja az ilyen esetek megelőzése, illetve a megtalálás megkönnyítése. Az internetet böngészve egy több mint tíz évvel ezelőtt megjelent rövid cikkre bukkantunk, mely szerint a jövőben az állatokat számítógép segítségével fogják őrizni, és még az egészségügyi állapotukat is így fogják figyelemmel kísérni. Ez szolgáltatta munkánk alapötletét.

2. Probléma megoldásának menete

Megoldásként egy olyan eszközt fejlesztünk, mely képes GPS adatokat szolgáltatni egy szervernek, melyeket a felhasználó a számítógépe segítségével megtekinthet, valamint SMS értesítést küld, ha a koordinátái a megadott területen kívülre kerültek. A fejlesztést két részre osztottuk, a hardver megépítésére és programozására, valamint a webes kezelőfelület programozására.

2.1. Hardver és működése

A hardver előre gyártott elemekből épül fel. A Mikroelektronika cég által gyártott Ready for PIC fejlesztőpanelt (1. ábra) valamint GPS és GSM/GPRS modulokat, úgynevezett Click Boardokat használtunk. Mivel a panel gyárilag nem kompatibilis ezekkel a modulokkal, némi módosítást végeztünk rajta. Munkánk során törekedtünk az eszköz méretének csökkentésére. A jelenleginél kisebb méretű megvalósításához saját hardver gyártása lenne szükséges, ám ez sajnos jelenleg nem áll módunkban.

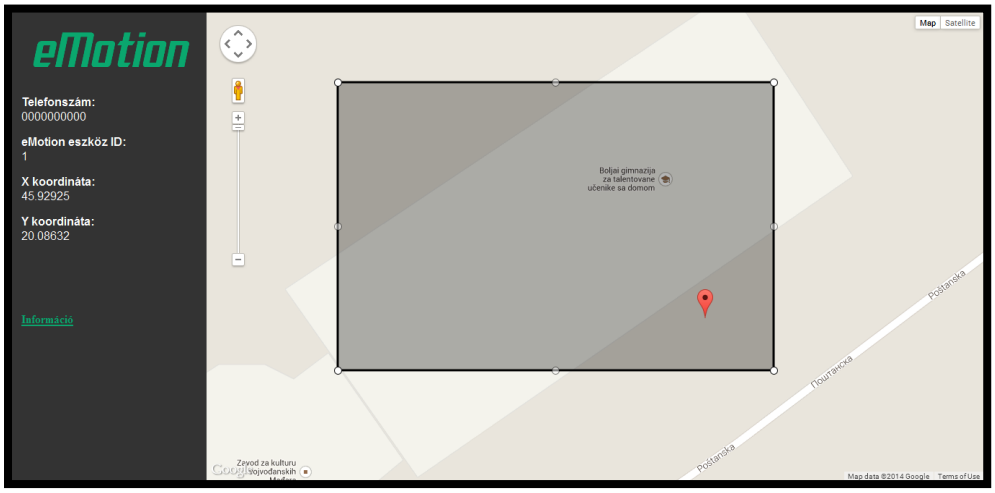
1. ábra. Ready for PIC fejlesztőpanel

A hardver szíve egy PIC18F45K22 típusú mikrokontroller, melyen az általunk írt program fut. A program a Click Boardok vezérlését, valamint azok adatainak feldolgozását végzi. Az eszközök közötti kommunikáció soros UART protokoll szerint történik. A GPS modul úgynevezett NMEA(*National Marine Electronics Association*) protokoll szerinti mondatok formájában küldi az adatokat, melynek $GPGGA mondatából kerülnek kiolvasásra az eszköz aktuális koordinátái. Például:

$GPGGA,183254.000,**4555.7550,N,02005.1795,E**,1,10,1.0,102.8,M,39.5,M,,0000*5A

A feldolgozás után a GSM/GPRS modul segítségével feltöltésre kerülnek a szerverre a koordináták, valamint az eszköz letölti a megadott terület határainak koordinátáit. Ha az aktuális pozíció a határokon kívül esik, SMS értesítést küld az előre megadott telefonszámra. Ezt a modult a mikrokontroller AT parancsok segítségével irányítja.

2.2. Webes felület



2. ábra. eMotion webes felület

Az általunk készített webes felületen (2. ábra) nyomon követhető az eszköz mozgása, illetve módosíthatóak a hozzá tartozó beállítások. A HTML, JavaScript és PHP technológiák felhasználásával készült kezelőfelület a Google Maps szolgáltatás térképeit használja fel a helyzet kijelzésére, valamint a megengedett mozgásterület határainak kijelölésére. Az eszköz a koordinátákat és a határokat egy SQL adatbázisban tárolja, és változások esetén frissíti azokat. A weboldal kialakítása során törekedtünk arra, hogy egyszerű, felhasználóbarát felületet alakítsunk ki, valamint AJAX technológiák használatával elértük azt, hogy az oldal frissítés nélkül is képes legyen megjeleníteni az eszköz mozgását, illetve módosítani az engedélyezett mozgásterület határait

3. Elért eredmények

Munkánk során sikerült olcsó, alacsony áramfogyasztású, és az elvégzendő feladathoz megfelelő eszközt létrehoznunk. Ennek ellenére tovább folytatjuk a fejlesztést, hiszen a mikrokontroller szabad erőforrásai ezt megengedik. A jövőben biofizikai mérések végzésére szeretnénk alkalmassá tenni az eszközt, mellyel így megvalósulna a valós idejű egészségügyi megfigyelés is. Továbbá igyekezni fogunk az eszköz méretét és árát tovább csökkenteni.

Háromdimenziós led mátrix

ESZI szakkör

Papp Zoltán, Tusa Balázs, Tóth-Albert Tibor
Energetikai Szakközépiskola és Kollégium, 7030 Paks,Dózsa György út 95.

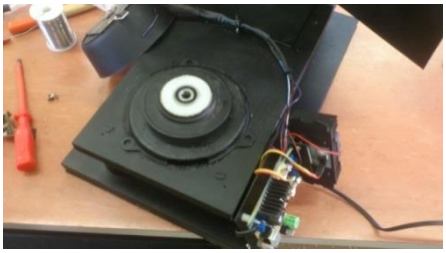
1. Bevezetés

Az Energetikai Szakközépiskolának az elektronikai szakmacsoportos tanulói vagyunk. Iskolánkban már hagyomány, hogy évente több alkalommal szakmai bemutatók keretén belül népszerűsítjük az iskolát a hozzánk ellátogató vendégek előtt. Folyamatosan építünk olyan áramköröket és berendezéseket, melyek látványosan tudják bemutatni, hogy mivel foglalkozunk, mit tanulunk itt az iskolában. Azt szeretnénk elérni, hogy az itt végzett munkáink látványosságukkal felkeltsék az emberek figyelmét, a pályaválasztás előtt álló diákokat pedig elektronika szakma felé csábítsa. Mivel az elmúlt tanévben már építettünk egy hasonló működési elven alapuló áramkört ezért már akkor elhatároztuk, hogy csinálunk egy 3D-s LED mátrixot, amely különböző animációkat jelenít meg a térben. Ennek a feladatnak a megoldása a mechanikai és elektronikai tevékenység mellett igen nagy programozási alkalmazást tett szükségessé, melyhez szakmai tanárunk sok segítséget és szakmai támogatást biztosított.

2. Problémamegoldás menete

2.1. Mechanikai felépítés

A LED mátrixot a mechanikai szerkezet megépítésével kezdtük el legyártani. Alaposan megterveztük, hogy hogyan és milyen szerkezeti elemekből állítható össze. A berendezés működés közben nagy sebességű forgómozgást végez, mely a szerkezet minden elemére rendkívül nagy igénybevételt fejt ki. Az alapos átgondolás ellenére az első verzió igen csak kezdetlegesnek bizonyult. A forgó mozgás következtében fellépő rázkódás volt a legfőbb probléma. A szerkezet stabilabbá tétele érdekében a forgó részeket csapágyaztuk, a villanymotor és a forgótengely közé hajlékony tengelykapcsolót építettünk. Ezt mutatja be az első ábra. Ezen javítóintézkedések hatására a szerkezet stabilitása lényegesen jobb lett, mely így biztos alapot tesz lehetővé az elektronikai elemek beépítéséhez.



1. ábra. Villanymotor a tartóállványban

2.2. Elektronikai felépítés

Az álló és a forgó alkatrészek között a tápfeszültség közvetítését csúszógyűrűkkel és szénkefék alkalmazásával oldottuk meg. A következő feladat a nyomtatott áramköri lap megtervezése volt. Az áramkőr alapját 4 darab AVR ATMEGA 128 mikro-vezérlő adja, melyek egymással master-slave üzemmódban működnek. Figyelembe kellett venni a felvillanó ledek számát, milyen szolgáltatásokat legyen képes nyújtani és mindez beépíthető legyen egy forgó szerkezetbe. Az áramköri lap Sprint-layaout programmal lett tervezve. A minél kisebb megvalósítás érdekében SMD technológiát alkalmaztunk. A 4db mikro-vezérlő kivételével az összes alkatrészt mi építettük be az áramkörbe. Szakmai tanárunktól sok segítséget kaptunk a tervezésben, megépítésben és a kész áramkör tesztelésében. Ez után a ledek beépítése következett, melyeket keménypapírból készített körív felületére építettünk. Ezzel a megoldással egyszerre biztosítottuk az RGB ledek lábai között a szigetelést és a könnyű szerkezet megvalósítását. Végül a villanymotor vezérlő áramkőrének a megépítése következett. Itt figyelembe kellett venni, hogy a motor fordulatszám szabályozására lesz szükség, ezért úgy döntöttünk, hogy PWM üzemmódban kell működtetnünk a villanymotort. Ennek megvalósításához egy PIC16F628 mikro-vezérlőt alkalmaztunk, mely egy BUZ11 MOSFET tranzisztorral kapcsolja a tápfeszültséget a villanymotor tekercseire. A tápegység áramkőrnek a

feladata, hogy biztosítsa a villanymotor és a mikro-vezérlők számára a 12V feszültséget. Az 5V-os feszültségeket pedig minden áramkörön lokálisan oldottuk meg. A kezelőszervekkel egybeépített tápegységet pedig a mechanikai szerkezet oldalára szereltük fel.

2.3. Programozás megvalósítása

A programozás megkezdéséhez először szét kellett osztani a feladatokat a processzorok között. Az 1-es mikro-vezérlő látja el a master szerepét. Ő állítja elő a különböző szolgáltatásokhoz szükséges adatokat, az animációk körül futó szövegek kiíratását és biztosítja az engedélyezéseket a többi processzor számára. A 2-es, 3-as, 4-es processzorok között az RGB színek működtetése lett felosztva. A képek megjelenéséhez a szerkezet egy fordulatának ideje alatt 123 variációban kell kigyújtani a megfelelő ledeket. A ledek felvillanását egy 8 bites rendszerben osztottuk szét. Az eltérő színek megvalósítását színkeveréssel oldottuk meg. A PIC16F628 mikro-vezérlő programozását assembly programnyelven MPLAB alkalmazásával készítettük, az AVR-ek programozása pedig Basic programnyelven BASCOM-AVR segítségével kerültek megvalósításra. A karakterek megjelenítésére az 5/7-es képalkotást alkalmaztuk. A szövegek bevitelére ASCII-kódok alkalmazásával van lehetőség. Maximum 192 darab karakterhosszúságú szöveg írható be, és ebből egy időben mindig 32 karakter jelenik meg. A mikro-vezérlő és a DS1307 Real Time Clock IC között I^2C buszos, a DS1821 hőmérséklet IC között pedig onewire kommunikáció megvalósítását alkalmaztuk. Az animációk képeit kinyomtattuk és felosztottuk 123 részre. A 40 db led-ből 5-ször 8 bites csoportokat hoztunk létre, s ez által képet képpontokká alakítottuk. A programozásban ezeket a képpontokat jelenítjük meg mikro-szekundomos időeltolással és így a szem tehetetlensége miatt egy összefüggő kép jelenik meg előttünk. Ezt a képet mutatja be a második ábra.

2.ábra. Háromdimenziós látvány

3. Elért eredmények

A projekt megvalósítása során olyan szakmai ismeretekre, tapasztalatokra tettünk szert, amikkel iskolai foglalkozás keretében talán nem is találkozhatunk. Magas szinten kiegészítette a már meglévő tudásunkat, és széles körben kiterjesztette a rálátásunkat azokra a technikákra, ahol az élet bármely területén alkalmazhatóak különböző feladatok, problémák megoldásában. A szerkezetet már kiállítottuk több helyen is az ősz folyamán ahol pályaválasztási tanácsokat hallgattak a meghívott vendégek. Örömmel tapasztaltuk, hogy nagy figyelmet és érdeklődést váltott ki az emberekben a látvány, a kitűzött céljainkat maximálisan elértük. Reméljük, hogy sok pályaválasztás előtt álló tanulónak az érdeklődését is sikerült felkelteni az elektronika tudománya iránt. Bízunk benne, hogy a jövőbeli munkáink során is képesek leszünk alkalmazni az itt szerzett ismereteket és hasznos szereplőivé tudunk válni a társadalom számára.

Pókerező Robot

Hegyesi Donát

Hegyesi Donát Sándor
Tóth Árpád Gimnázium, 4024 Debrecen, Szombathi István u. 12.

1. Bevezetés

Gyerekkorom óta foglalkoztat a gépek, a robotok működése, építése, programozása. 14-15 évesen ismerkedtem meg közelebbről a programozással és a programozható áramkörökkel. Egy robot építése mindig is nagy álmom volt. 2012 nyarán született meg a pókerező robot gondolata, miközben pókert játszottam a barátaimmal. Úgy véltem ezzel a robot megépítésével megvalósíthatnám az álmomat.

Egy pókerjátékost helyettesítő gépet építettem, ami az emberekhez hasonlóan képes felismerni a lapokat, zsetonokat mozgatni és döntést hozni bármilyen szituációban.

2. Probléma megoldásának menete

Egy 4 szabadságfokkal rendelkező manipulátor építésével kezdtem a munkát. Minden elemét számítógéppel terveztem és forgácsolással vagy lézeres megmunkálással gyártattam le, a lehető legnagyobb pontosság elérése érdekében. Ezzel a manipulátorral a robot képes mozgatni a zsetonokat (ily módon tud dobni, passzolni, megadni és emelni). A robot vezérlőegysége egy ipari számítógépre épül. Linux operációs rendszer alatt futnak a különböző feladatok ellátását biztosító programok.

A gépi látáson alapuló kártyalap felismerést találtam a legjobb megoldásnak az ember-gép kapcsolat egyik megvalósítására. Ennek megfelelően a robot két kamerával érzékeli az ellenfél reakcióit és a képek elemzésével azonosítja az egyes lapokat, zsetonokat.

Az általam megírt pókerező algoritmus a döntéshozatalnál figyelembe veszi a saját és a közös lapokat, az ellenfél lépését, a robot és az ellenfél zsetonjainak számát, illetve a közös kassza méretét. Ezeknek az adatoknak az összességéből próbálja meghozni a legoptimálisabb döntést. Az összetett döntési folyamat modellezésére fuzzy logikán alapuló algoritmust alkalmazok.

2.1. Az első prototípus

Több mint 9 hónapot vett igénybe az első prototípusom megépítése, ami az 1. ábra látható. Minden elemét kézzel gyártottam le. A robotkarhoz 3mm-es rétegelt lemezt választottam és standard méretű analóg szervomotort. Az első verziónál a célom a saját pókerstílusom modellezése volt. Az elkészítést követően, rögtön jelentkezett egy probléma: túlságosan könnyen ki tudom ismerni a robotot, hisz hasonlóképpen gondolkozik, mint én. E miatt döntöttem a pókerező robotom fejlesztése mellett.

1. ábra. Az első prototípus

2.2. A második prototípus

A 2. ábrán látható a robotom, ami nem csak szoftveresen, hanem hardveresen is javult. A robotkar elemeit immáron lézerszabászatnál gyártattam le, a lehető legnagyobb pontosság elérése érdekében. Az analóg motorokat erősebb és pontosabb digitális szervomotorokra cseréltem. A nagyobb játékélményért egy érintőkijelzőt is szereltem a pókerező robotra. Erre a kijelzőre fejlesztettem egy grafikus programot, amelyen 3 nyomógomb segítségével az egész játékot irányítani lehet. A robot döntési algoritmusa a fuzzy logikára épül, így sok paraméter figyelembevételével, képes optimális döntést hozni. Időnként még blöfföl is.

2. ábra. A második prototípus

3. Elért eredmények

A pókerező robotomon már több mint másfél éve dolgozom. Véleményem szerint egy tapasztaltabb pókerjátékos ellen is megállná a helyét. Alapvetően kétféle területen tudom elképzelni ezt a robotot. Egyrészt a kaszinókban, mint játékgép. Új színt hozna a termek életébe, hiszen a pókerben nincs 100%-ig biztos nyerő stratégia, így ez egyfajta igazságos játék lenne. Másfelől pedig a fejlődni kívánó játékosok használhatnák, mint gyakorlópartnert.

Légbőlkapott

Légbőlkapott

Vincze Zoltán Dániel
Szent Imre Általános Iskola, Gimnázium és Szakközépiskola, 2500 Esztergom, Főapát utca 1.

1. Bevezetés

A Hexa-kopter építésének célja légi felvételek készítése olyan helyeken, ahol normál repülővel vagy helikopterrel nem megközelíthető a helyszín. Például árvizek estén a kiépített gátak ellenőrzése, videó felvételek készítése szűk helyeken, csarnokokban, épületek között, régészeti feltárásokhoz koordináták alapján fényképek készítése a kutatás segítése érdekében. E mellett csomagok emberi beavatkozás nélkül, kijelölt célhoz való eljuttatására is alkalmazható.

2. Probléma megoldásának menete

Cél az volt, hogy legalább 4 kilogramm hasznos terhet fel tudjon emelni, hisz egy jobb minőségű fényképezőgép 2 kilogramm körül van.

A gép 4,6 kilogrammos önsúlya mellé 12 kilogramm tolóerő társul, ami növelhető nagyobb rotorokkal. A felhasznált alkatrészek nagy része nem megvásárolható, így ezeket megterveztem és elkészítettem. Azért választottam a hat rotoros kivitelt, mert esetleges motorhiba esetén 5 motorral a gép irányítható marad és így kisebb az esély a lezuhanásra. A váz anyaga szénszál, illetve alumínium. Energiáról 2db 5000mAh-s, 6 cellás lítium akkumulátor gondoskodik. Meghajtómotoroknak 3 fázisú motorokat választottam, amiknek darabonkénti csúcsteljesítménye megközelítőleg 600Watt. A repülésszabályozó rendszer az Arduino MEGA fejlesztőpanel alapjára épül, a szenzorok nagy része integrálva van egy panelra, kivétel a GPS, az iránytű és a feszültség-árammérő modulok.

Az első ábrán az összeállítás forrasztása látható. A vázat többször szétszedtem. Először elektronika nélkül lett összerakva, hogy láthassam hogyan helyezhetném el megfelelően a különböző szenzorokat. Másodjára a csavarok fixálása miatt lett elemeire bontva.

1. ábra A kopter összeszerelés közben

2. ábra Repülés közben

A gép biztonságos működéséről több egymástól független rendszer gondoskodik. Első ilyen a repülésirányító programban van. Ha bármilyen rádió kimaradás, zavar keletkezne, a gép felemelkedik egy előre beállított biztonsági magasságba, a felszállási „home"pont felé fordul, hazatér és landol. Az akkumulátorokat többféleképpen monitorozom. Első a repülésirányító feszültség és árammérő szenzor, ami egy kritikus pont elérésekor a multikoptert hazahozza. A második az akkumulátorokon elhelyezett feszültség figyelő, ami hangjelzést ad, ha bármely cellafeszültség beesik a minimum alá. Harmadik a távirányító kijelzőjére kiírja az akkuk kapocsfeszültségét és a távirányítóban beállított alsó határnál hangjelzést ad. A negyedikként a gép állandó kapcsolatban van egy Android alapú tablettal, ami hangosan kimondja az akku tartalékot százalékban, továbbá tájékoztat a magasságról, repülési módról, valamint itt megtervezhetem az önálló repülést egy térképen. Az biztonsági rendszer ötödik elemei az RGB led szalagok, amelyeket az ANRO Épületdíszítés Kft.-től kaptam. A második ábrán látható. Ezek irányfényként működnek. A ledek villogása visszajelzést ad a GPS kapcsolatról, valamint az akkumulátorok állapotáról.

A rádiós kommunikáció 3 frekvenciatartományban történik 2,4GHz-en a manuális irányítás, 433Mhz-en a gép telemetriás adatai és 5,8Ghz-en az élő videó kép. 433Mhz-re és 5,8Ghz-re saját készítésű antennákat használok.

3. Elért eredmények

A tervezéssel nem volt különösebb gondom. Egy hónap alatt kiforrott a konstrukció. Minden alkatrészt elsőként papíron rajzoltam meg. A tervezés érdekében megtanultam kezelni a CorelDRAW Graphics nevezetű programot és az alkatrészek egy részét ezzel készítettem el. Nehézséget okozott az alumínium alkatrészek precíz megmunkálása, de hála egy gépész ismerősömnek, megismertem egy marógép kezelését, így az alkatrészeket nagy pontossággal tudtam elkészíteni. Körülbelül 90db egyforma bilincset készítettem el a karbon csövek rögzítéséhez. A NOVIA Kft.-től kaptam szénszövetet, amit laminálni kellett. A végleges alkatrészek elkészítését 3 kísérlet előzte meg. Körülbelül 5 nap küzdelmes munka után elkészültek a karbon elemek a multikopterhez. A komplett szerkezetet 1-2 hét alatt raktam össze 214db csavar felhasználásával. A gép P.I.D. beállítása körülbelül 2 hét alatt napi 40 perc aktív repülés után lett elfogadható. A szerkezettel több alkalommal az esztergomi repülőtéren repültem. A repülések során teszteltem a gép autonóm funkcióit. Például az automatikus hazatérést és a koordinátákra repülést. Műterhelési teszten 5,25kg súlyt emelt a magasba. Ekkor kiderült, hogy a leszállótalpak nem bírják a terhelést, elhajlanak. Ebből a következtetésből arra jutottam, hogy masszívabb, nagyobb teherbírású anyagból kell kiviteleznem. A tesztek során másik problémára is fény derült. A karok végeit össze kell kötnöm a rezonancia elkerülése végett. Összességében jól teljesített a hexa-kopter. A hibák orvoslása után a kitűzött célt, a várt eredményt teljesítette.

A XXI. század menekítő csapata

LegoRockers

Bálint Karola, Gilicze Kristóf, Sebők Attila
Csongrádi Batsányi János Gimnázium, Szakképző Iskola és Kollégium, 6640 Csongrád, Kossuth tér 1.

1. Bevezetés

Tegyük fel, hogy valahol valamilyen katasztrófa történt, mondjuk egy földrengés következtében megsérült egy ház, vagy egész városrész, ipari létesítmény esetleg egy atomerőmű. Ha fontos ezeknek a területeknek az átjárhatósága (pl. stratégiai útvonal, utánpótlás szállítás), akkor a megváltozott terepviszonyok között új útvonalat kell keresni.

Ezt a feladatot robotokkal célszerű megvalósítani, mert így nem kockáztatunk emberéletet. A cél az volt, hogy gyorsan rövid útvonalat találjunk, ami a katasztrófa sújtotta területen átvezet. A mi megoldási ötletünk arra épül, hogy több robot hatékonyabban oldja meg ezt a problémát. A teszteléshez készítettünk egy modellt, ahol a terepet labirintus formájában ábrázoltuk. A robotok egymás között bluetooth-szal kommunikálnak.

2. Probléma megoldásának menete

2.1. A labirintus

A labirintus derékszögű, egyforma falapokból lett építve egy bejárattal és egy kijárattal, ami az 1. ábra középső képén jól látható. .Az elrendezése tetszőlegesen variálható, a labirintuscellák közepén egy fekete vonal segíti a robot közlekedését. Ez a modell első megközelítése a valóságos helyzetnek.

2.2. A robotok

Három robot (egy master, két slave) kommunikál egymással. A két slave robot felépítése (1. ábra, baloldali fotó) megegyezik: kisméretűek, mozgékonyak, három ultrahangos távolságérzékelővel és egy fényérzékelővel tájékozódnak.

A masternek (1. ábra, jobboldali fotó) nincs szüksége távolságérzékelőre, mert a slave-ek által küldött információk alapján határozza meg saját útvonalát.

1. ábra. A slave robot, a labirintus és a master robot

2.3. A stratégia

A két slave robot egymás után elindul a labirintusban, minden labirintusegységben megállnak, információt gyűjtenek környezetükről, ezt továbbítják a masternek. A master robot tárolja, kiértékeli, nyilvántartja a bejárt pozíciókat és a feldolgozott adatok alapján felváltva továbbhaladási utasítást küld a két slave robotnak. Az egyik robot esetén a jobb oldalt, a másiknál a bal oldalt részesíti előnyben. Miután mindkét slave robot kijut a labirintusból, a master kiválasztja a rövidebb útvonalat, és ezt követve, de az esetleges zsákutcákat kihagyva, áthalad a labirintuson.

Többféle optimalizálása létezik a problémának: a legrövidebb úton, vagy a lehető leghamarabb átjutni a labirintuson.

2.4. Az algoritmus

A labirintus egységeket koordináta párokkal (x;y) azonosítjuk. Ezek értéke segít meghatározni az aktuális pozíciót és a labirintus végét.

A slave robot minden labirintusegységben megvizsgálja az adott pozícióból a továbbhaladási lehetőségeket. Az utat 1-essel, a falat pedig 0-val jelzi. Ezt a kódhármast küldi a masternek, mely válaszként egy karaktert (b,e,j,h) küld. A slave robot ennek megfelelően balra, előre, vagy jobbra megy. A h egy zsákutcát jelöl, ilyenkor a slave megfordul, és előre megy. Lényegében ez ismétlődik, amíg a slave ki nem jut a labirintusból.

A master kiválasztja az optimalizált útvonalat, majd a következő zsákutcakezelést alkalmazza: minden pozíciót megvizsgál, hogy szerepel-e még egyszer a tervezett útvonalban. Ha szerepel, akkor a pozíció utolsó előfordulására ugrik.

2.5. Adatszerkezet

Az adatok tárolásához két kétdimenziós tömböt használtunk. Egyikben a jobbra tartó slave, a másikban a balra tartó slave robot által küldött, valamint a master robot által feldolgozott információk találhatók, ahogyan ez az 1. számú táblázatban látható.

X	**Y**	Falak	Mozgás
1	1	000	J
1	2	101	E
1	3	110	B
2	3	111	H

1. táblázat. Koordináták és utasítások

2.6. A programozás

A programokat RobotC nyelvben fejlesztettük. Összesen két program szükséges, az egyik a masteren fut, a másik a slave roboton (mindkettőn ugyanaz a programkód).

2.7. Problémák

Kezdetben NXTG-ben programoztunk, de végül „zsákutcába" jutott fejlesztésünk, mert a file-ban eltárolt adatokat körülményesen tudtuk kezelni. Ekkor döntöttük el, hogy RobotC-ben írjuk meg a programot.

Ebben a fejlesztő környezetben a bluetooth kommunikációt kettő robot között is nehéz volt összeegyeztetni, a harmadik robot pedig már igazi kihívást jelentett.

A külső körülmények (megvilágítás, ragasztott pályaelemek, súrlódás) nehezítették az algoritmus tesztelését. Végül ezt is sikerült megoldanunk.

3. Elért eredmények

A labirintus-bejárási feladatok másfél-két éve kerültek a látóterünkbe, ez idő alatt többféle szempontból megközelítettük a problémát:

- Megismerkedtünk tisztán elméleti bejárási stratégiákkal (rekurzív, és nem rekurzív megoldások).
- Foglalkoztunk teljes labirintus bejárással, és a bejárathoz visszajutó a robot algoritmusával.
- A gyakorlatban történő teszteléshez mindig robotokat használtunk.
- Többféle méretű labirintusban, több robotkonstrukcióval valósítottuk meg a fenti algoritmusokat.

Az ilyen képességű robotok, a katasztrófahelyzetekben alkalmazott un. menekítő robotok, amelyek vészhelyzetben emberi életet, és egyéb értéket menthetnek, szerepet kaphatnak a szennyezett területek mentesítésében is.

Az általunk megírt algoritmust továbbfejlesztve egy olyan menekítő csapatot lehetne létrehozni, mely hatékonyan és gyorsan működik. Kettő, vagy akár több robot feltérképezi a területet, megtalálja a legoptimálisabb útvonalat, majd egy további robot ezen az útvonalon egyszerűen végighalad, és megteszi a szükséges intézkedéseket. Ez lényegesen gyorsabb és biztonságosabb, mint ha minden emberekre lenne bízva.

Róka-nyúl játékgép

Materex

Kiss Máté
Kecskeméti Bányai Júlia Gimnázium, 6000 Kecskemét, Nyíri u. 11.

1. Bevezetés

A projektben egy olyan MINDSTORMS NXT programozható mini komputerekre épülő konstrukció megépítése és programozása volt a cél, amely képes egy sakktáblán játszható egyszerű játék modellezésére. A játék kétszemélyes. A megépített konstrukció lehetővé teszi, hogy a két játékos az NXT alapú konzolok segítségével a valóságban is megtegye a lépéseit és ne csak a képernyőn jelenjen meg a lépés szimulációja. Ezzel a céllal tehát a konzolos játékok helyett egy sokkal látványosabb játék szimulációt valósítottunk meg. Kiegészítésként az egyik játékos szerepét átveheti a gép, így egy „mesterséges intelligencia" ellen is lehet játszani, bár a gép nem lesz intelligens, de a mögöttes algoritmusok a külső szemlélő számára ezt a benyomást nyújtják.

2. Probléma megoldásának menete

A fő feladatok tehát a projekt megvalósítása során:

- a modellezett játék kiválasztása,
- konstrukciós szempontból a bábuk mozgatásának megoldása a játékosok konzolon beállított lépéseinek megfelelően,
- programozási szempontból legalább három NXT komputer közötti kommunikációs program elkészítése,
- valamint a számítógép önálló játékát lehetővé tevő egyszerű algoritmus tervezése és kivitelezése.

A „mesterséges intelligencia" algoritmus, amely a konstrukció önálló döntéshozatalát végzi további fejlesztést igényel, jelenleg egyszerű, néhány szempontú paraméterlistával működik.

2.1. A játék

A kiválasztott és modellezett játékként próbáltunk olyat keresni, amelyben a két játékos közül egyiknek sincs nyerő stratégiája, sakktáblán játszható, és viszonylag kevés számú bábut tartalmaz.

A kiválasztott játék egy régi lengyel népi játék egy változata, amely a két játékos szempontjából aszimmetrikus, hiszen sem a bábuszám, sem a lépési lehetőségek nem azonosak, de a kiválasztott kritériumok mindegyikét teljesíti. A játék neve: róka-nyúl.

- A 8x8-as sakktáblán játszható, 4 róka bábu és 1 nyúl bábu segítségével.
- A rókák a sakktábla egyik szélén lévő sor négy fehér mezőjén helyezkednek el, míg a nyúl a rókákkal átellenes sor valamelyik középső fehér mezőjén.
- Minden bábú átlósan léphet a szomszédos mezőkre egyet. A rókák csak előre (két irányba), míg a nyúl előre és hátra is (négy irányba). A két játékos a rókákat illetve a nyulat irányítja és felváltva lépnek. A bábukat nem lehet átugrani és ütés nincs.
- A rókák célja a nyúl bekerítése, tehát olyan állás kialakítása, melyben a nyúl nem tud lépni. A nyúl célja a rókák támadó vonalán történő átjutás.

2.2. A konstrukció

A konstrukció elkészítéséhez LEGO elemeket vettünk alapul, mivel ezek gyakorlatilag korlátlan variációs lehetőséget biztosítanak, egyszerű eszközökkel. Sakktáblaként egy készen kapható vékony préselt falemezt használtunk, amely eleve ezt a célt szolgálta. A bábuk mozgatására több különböző lehetőséget is kipróbálva a legideálisabb megoldásként egy mágnest választottunk, amely megfelelő távolságból képes az állandó mágneses mező segítségével a falemezen keresztül is mozgatni a bábukat, amelyek természetesen mágnesezhetők kell hogy legyenek. Így nincsenek a tábla fölött a láthatóságot befolyásoló alkatrészek. A sakktábla vastagsága ugyanakkor lehetővé teszi a mágnes árnyékolását, így a tábla alatt változtatva a mágnes távolságát az egyes bábuk egyedileg megfoghatók, mozgathatók és elengedhetők.

A játékhoz három különálló konstrukciót kellett építeni: a két vezérlő konzolt és a sakktábla részt. A vezérlő konzolok felépítése hasonló, de mivel a két játékos (róka illetve nyúl) lépései nem szimmetrikusak, tehát részben különböző módon lépnek, ezért a két konzol is különbözik egymástól.

A sakktábla rész esetén ki kellett alakítani egy olyan szerkezetet, amely stabilan tartja a táblát és elegendő hely van a tábla alatt a pozícionálást végző szervo motoroknak, valamint a bábu „megfogásához” használt állandó mágnes le-fel mozgatásához. Mindezt úgy, hogy a pozíció megtalálásához szükséges mozgó keret ne akadjon el a tábla tartólábaiban. A bábuk mozgatása során szinte a teljes konstrukció mozog, így a tartószerkezetnek erősnek és több irányban rögzítettnek kell lennie, hogy a tábla ne remegjen, a bábuk ne mozduljanak el. A használt sakktábla nem illeszkedik a LEGO szabvány méreteihez, így ennek illesztése is külön feladat volt. Az elkészült konstrukcióról készült kép az 1. ábrán látható.

1. ábra. A játékhoz készült konstrukció (játéktábla és konzolok)

2.3. A program és a kommunikáció

Programozási szempontból legalább három darab MINDSTORMS NXT mini komputer szükséges a megvalósításhoz, hiszen a játékos konzolokhoz kell egy-egy, míg a bábuk mozgatásához, a motorok működtetéséhez kell a harmadik. A három eszköz közötti kommunikációhoz a bluetooth technikát használtuk, tehát az adatok bluetooth-on keresztül jutnak el a célhoz.

Hálózati szempontból egy master-slave elvű hálózatot építettünk, melyben van egy kitüntetett szerepű NXT tégla, amelyhez befutnak az adatok és ez továbbítja az információkat és vezérlő jeleket a másik kettőnek. Mivel nem akartunk egy negyedik mini komputert is használni, így masterként az egyik játékos konzolt választottuk.

A program forráskódja NXC programnyelven készült, amely egy C nyelvhez közeli magas szintű programnyelv és alkalmas a NXT mini komputerek programozására.

A programírás során a legfontosabb feladatok a tábla aktuális állásának tárolása, valamint a kiválasztott lépés megtételéhez szükséges koordináták motorvezérlőhöz történő eljuttatása volt. Adattárolási szempontból a két konzol komputere teljes körű információval rendelkezik az állásról, míg a motorvezérlő csak az aktuális mozgási pozíciót ismeri. Az adatküldési kódrendszert úgy alakítottuk ki, hogy egyetlen adatküldéssel lehessen a kommunikációban a régi és az új pozíciót átadni. A megkapott adatokból matematikai eljárással lehet a lépést dekódolni.

Az adatokon kívül a vezérlő jelek átadása is megoldandó programozási feladat volt. Például: az aktuális lépés befejezése a motorvezérlő részéről, a játék kiértékelése, lépésengedély, …

A használat során nem csak a táblán követhető nyomon a játék menete, hanem a mini komputerek pontmátrix kijelzőjén is. Lásd 2. ábra! Ezek programozása külön feladat volt.

2. ábra. A kijelzőn látható aktuális állás és lépési lehetőségek megjelenítése.

3. Elért eredmények

A kitűzött célt sikerült megvalósítani. Elkészült a konstrukció, amely képes két konzollal vezérelt kétszemélyes játék fizikai szimulálására, a játékosok lépéseinek egy sakktáblán való megjelenítésére. Mindehhez három NXT mini komputert használtunk, amelyek adatcserére képesek. A megvalósításhoz három szervo motort, két fényérzékelő szenzort, és öt ütközésérzékelőt használtunk. A hálózati kommunikációt bluetooth alapon valósítottuk meg. A program forráskódjai C-hez közeli NXC nyelven készültek. További lehetőségként építettük be a játékkonstrukcióba a számítógép elleni mérkőzést. A megtervezett és programozott algoritmus azonban kezdő szinten játszik (bár hiba nélkül).

Mérések mikrokontrollerrel napkollektor optimalizáláshoz

Murphy's Hands

Kiss Ádám, Szabó Benedek
Kecskeméti Bányai Júlia Gimnázium, 6000 Kecskemét, Nyíri u. 11.

1. Bevezetés

A technikai civilizáció XX-XXI. századi rohamos fejlődésével együtt jár az energiafelhasználás növekedése. A fenntartható fejlődés megőrzése érdekében a figyelem egyre inkább a megújuló energiaforrások és a hulladék újrahasznosítás felé fordul.

A célunk egy zömében újrahasznosított alapanyagokból (pl.: sörösdobozokból) készülő napkollektor megépítése volt. Az internetes szakirodalmi források véleménye megoszlik az ideális felépítés viszonylatában. Ezért olyan technikai kivitelezést és méréssorozatot terveztünk, amely alapján választ adhatunk az optimalizálás problémájára.

A mérések során vizsgáltuk az átáramló levegő sebességének, az átömlési profilnak, a dőlésszögnek, a külső és belső hőmérsékletnek a teljesítményre gyakorolt hatását. Mindehhez olyan moduláris konstrukcióra volt szükség, amely egyedi fejlesztésű. A mérések elvégzéséhez AVR-Duino mikrokontrollert és a kollektor csövekbe helyezett saját fejlesztésű hőmérő szenzorsort használtunk. Az informatikai hardver és szoftver tervezése és kivitelezése is saját munka.

2. Probléma megoldásának menete

Méréseink fontos része a napkollektor leadott teljesítményének vizsgálata. Mivel ez az érték több környezeti és belső változótól is függhet, el kellett érnünk, hogy a belső változók közül minél többre befolyásunk legyen, hogy megfelelően következtethessünk a teljesítmény egyes külső változóktól való függésére.

2.1. A konstrukció (röviden)

Napkollektorunkat az interneten föllelhető számos, korábbi verzióhoz hasonlóan egy fa keretbe beépítve készítettük el. A kollektorcsövek hátlap felőli oldalán kőzetgyapotos, hőtükrös szigetelést használtunk. Egy alumínium U szelvényből készített doboz szolgált gyűjtődobozként. A rendszer belső párásodását habszivacs alapú, gumis tömítő szigetelés gátolta.

A korábbiakban leírtak indokolttá tették egy olyan rendszer beépítését, amellyel a konstrukción belüli áramlási sebességet megfelelően pontosan lehetett szabályozni. Ezt a rendszert egy radiális ventilátorral (Maruti utastéri ventillátora) és az azt vezérlő áramkörrel oldottuk meg.

Az átáramlási sebesség, és közvetett módon a levegő mennyisége függ a kollektorcsövek belső légellenállásától is, amelyet a kivágási profil határoz meg. Ezért olyan dobozsorokat terveztünk, amelyek cserélhetők. Ezzel reprodukálható módon tudtuk vizsgálni a különböző átömlési profilok hatását.

Minden vezérlőáramköri elemet a túlmelegedés elkerülése érdekében a napkollektor házát adó fakereten kívül helyeztünk el. Az elkészült konstrukció fotóját lásd 1. ábra.

1. ábra. A napkollektor egy mérés során

2.2. Műszerezés

Mivel a belső áramlást biztosító ventillátorhoz csatlakoztatott motor 25 watt teljesítményű, hagyományos feszültségosztásos teljesítményszabályozásra nem volt lehetőségünk, így négyszögjeles áramlási sebesség vezérlést használtunk, amelyhez vezérlő áramkört készítettünk. Mivel a vezérlést megvalósító finomelektronikai eszközök nem képesek a kívánt teljesítmény leadására, ezek csak a szabályzójelet biztosítják, mellyel egy saját építésű áramkör állítja elő a

megfelelő kimenő feszültséget a motor számára (12V DC). Az így elkészített áramkör az induktív visszarúgásoktól is végi a mikrokontrollert. A négyszögjel kitöltési tényezője segítségével finoman állítható a motor teljesítménye (0-255 kitöltési tényező tartomány a 0-25 watt teljesítménytartományra). Az áramlási sebesség és a négyszögjel kitöltési tényező azonban nem lineárisan arányos a pl. a fellépő turbulenciák miatt.

A hőmérséklet gradiens meghatározásához 6 hőmérőt helyeztünk, egymástól egyenlő távolságokra a dobozsorban.

Az adatokat egy mikrokontrollerrel gyűjtöttük, amelyhez az adatrögzítő szoftver is saját fejlesztés. Az adattárolás a csatlakoztatott SD kártyán történt.

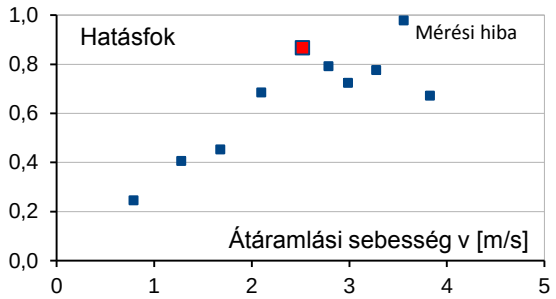
Méréseink során az egyik nehézség, a felszíni napállandó változása volt. Mivel ez az érték folyamatosan és jelentősen változik (egy mérés időtartamán belül pl.: 420-860 között), így ennek meghatározásához bolométert használtunk.

2.3. Mérések és eredmények (röviden)

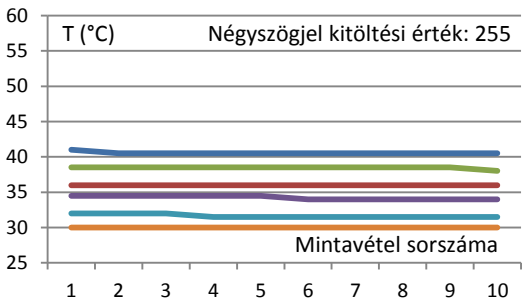
Első mérésként a leadott teljesítményfüggést vizsgáltuk a napsugarak napkollektorral bezárt szögétől függően, hogy leellenőrizzük szerkezetünk pontosságát. Kísérletileg bizonyítottuk, hogy a legnagyobb elnyelődés (és ezáltal a teljesítmény) akkor várható, mikor a napsugarak merőlegesen esnek be a szerkezet felületére.

Másik mérésünk során a rendszer hatásfokát vizsgáltuk a különböző átömlési profilok és átáramoltatott levegő mennyiségének függvényében. A hatásfokkal való összehasonlítás kedvezőbb, mivel az külső változóktól jóval függetlenebb, mint a leadott teljesítmény. Különböző sörösdoboz kivágási profilokkal mértünk. Minden kivágási profillal végigpróbáltuk a teljes lefedhető átáramlási sebességtartományt. A hatásfok maximumát kerestük az egyes csövek, és áramlási sebességek függvényében. Lásd 2. ábra. Két különböző profilú kollektorcsővel végeztük el a teljes méréssorozatot.

A szerkezetbe illesztett hőmérősorral mértük egyidejűleg a belső hőmérsékleteket dobozonként, ezzel megállapíthattuk a napkollektor hőmérséklet gradiensét. Így nem csak a beérkező és távozó energiát figyelhettük, hanem a szerkezet belsejében lejátszódó folyamatokra is következtethettünk, megállapíthattunk tendenciákat, ezzel is elősegítve az optimalizálást. Lásd például 3. ábra.



2. ábra. Hatásfok maximum



3. ábra. Hőmérsékleti gradiens

A napkollektor teljesítményét az (1) egyenlettel számoltuk. Az egyenletben szereplő mennyiségek a projekt mérési adataiból származnak, vagy belőlük számolhatók.

$$P_k = \frac{cm\Delta T}{\Delta t} = \frac{cA_{cs}v\Delta t\rho\Delta T}{\Delta t} = cA_{cs}v\rho\Delta T \qquad \textbf{(1)}$$

3. Elért eredmények

Célunk nem egy általános érvényű építési útmutató elkészítése volt, hanem segítségnyújtás, ötletadás azon emberek számára, akik maguk is napkollektorépítésre vállalkoznak. Lehetőséget teremtve ezzel, akár kisebb helyiségek fűtésére és a megújuló energia egyszerű felhasználására.

Megállapítottuk, hogy minél nagyobb egy kollektorcső belső légellenállása, annál hatékonyabbnak látszik az áramlási sebesség csőhöz igazítása után. Sikerült egy olcsó, kivitelezhető megoldást találnunk a levegőmozgató rendszer felépítésére is.

A hőmérsékletgradiensek esetében adódott, hogy minél több levegőt áramoltatunk át a kollektoron, annál közelebb kerülnek egymáshoz az egyes kollektorcsöveken belüli dobozokban mért hőmérsékleti értékek. Tehát átömlési profilonként és áramlási sebességenként létezik egy optimális csőhossz, ahol az újabb dobozok hozzáadása már nem növeli szignifikánsan a teljesítményt.

Reboss csapat

Reboss csapat

Bosznai Tamás, Remete Bence
Beszédes József MMIK, Magyarkanizsa Széles utca 70.

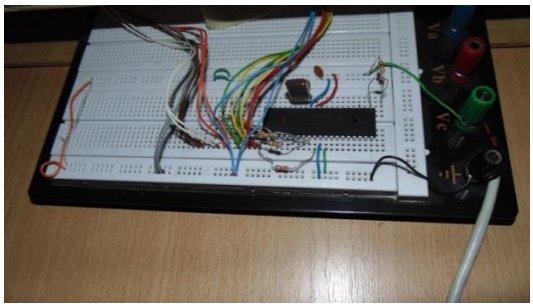
1. Bevezetés

Egyik órán a tanár úr meg említette, hogy lesz egy verseny tőlünk nem messze Szegeden és, ha valakinek van, valami ötlete az jelentkezzen. Padtársammal rögtön egymásra néztünk, innentől nem volt kérdés, hogy megpróbálunk-e összedobni valamit. Sokat keresgéltünk mindketten a neten és különböző könyvekben, mert nem tudtuk, hogy milyen témát is válasszunk. Végül az ötletet egy hobbielektronika nevű weboldal adta, ahol olvastam egy cikket „Nulláról a robotokig" címmel, amely nagyon megtetszett. Ez a cikk bemutatja, hogy egy PIC segítségével hogyan lehet kis LED égőket különböző módon villogtatni. Mivel településünkön épp most építettek két helyen is villanyrendőrt, ezért jött az ötlet, hogy meg kellene csinálni az útkereszteződés kis makettjét rendesen működő lámpákkal.

2. Probléma megoldásának menete

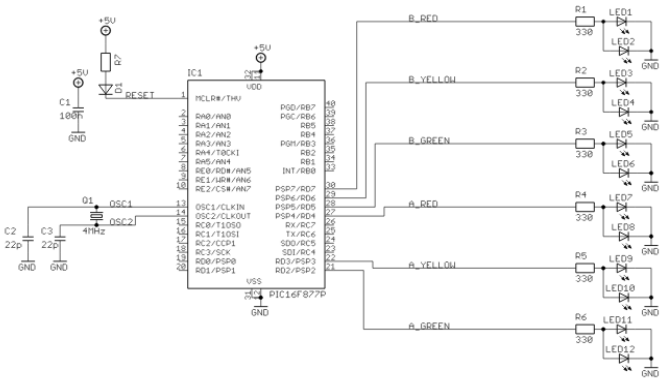
2.1. Programozás

Szakemberhez fordultunk csapattársammal, hogy kiderítsük hogyan is kell működnie egy szabványos villanyrendőrnek. A hallottak alapját összeállítottunk egy algoritmust. Ezután pedig maga az áramkör megépítése következett – az 1. ábra látható.



2. ábra Megépített áramkör

Alapul egy próbapanelt használtunk, amire rátettük a PIC -et a LED égőket, ellenállásokat, oszcillátort, kondenzátorokat és diódákat. A cikkben és az interneten találtunk elég sok kapcsolási rajzot a PIC -hez, amit mi is szerettünk volna használni így ezekből összeállítottuk a nekünk legjobban megfelelőt. Ez látható a 2. ábrán.



3. ábra Kapcsolási rajz

Jöhetett a program megírása. A programot MPLAB IDE programban C++ programnyelven írtuk meg.

2.2. Makett építés

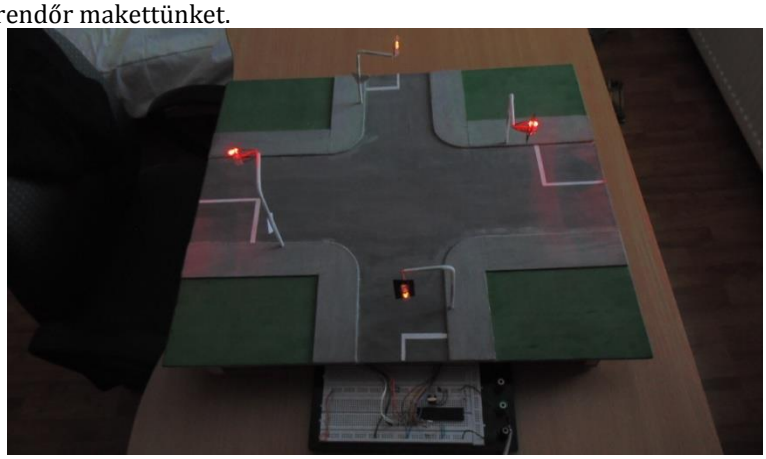
Amikor már megvolt minden és a próbapanelen működött, elkezdtük megtervezni a makettet. Az alap egy préselt lemez, amelyre furnérlemezekből vágott darabokat ragasztottunk, ezek jelölik a járdát. A villanyrendőr tartóoszlopait pedig szívószálakból készítettük el. A próbapanelen csak 6 db LED volt, de a maketten 12 darabnak kell lennie- a 3. ábrán látható.

4. ábra Kész makett

Ezt úgy oldottuk meg, hogy a vezetékek is elférjenek a kis szívószálban, hogy az egy lámpához tartozó 3 kis LED földelését egybe kötöttük, valamint a próbapanelből minden kimenetnél két vezetéket egybekötöttünk így a két szemben álló egyformán működő lámpák problémáját is letudtuk. Így elkészült az útkereszteződés.

3. Elért eredmények

Sikeresen megvalósítottunk egy működő villanyrendőrrel ellátott útkereszteződést. A mindennapi életben is számos alkalommal találkozunk ilyen rendszerekkel. Sikerült elmélyedni a C++ programnyelvben és felfedezni a Micro chipek világát. Ez feltárt előttünk számos új lehetőséget, amit majd a későbbi munkáink során szeretnénk kamatoztatni. Végül bemutatnánk a 4. ábrán a működő villanyrendőr makettünket.



5. ábra A működő villanyrendőr

A Tóth Árpád Gimnázium új csengője, a tAGBell

TeamTAG

Alekszejenkó Levente, Ugrin Dániel
Tóth Árpád Gimnázium; 4024, Debrecen, Szombathi István u. 12.

1. Bevezetés

A *Tóth Árpád Gimnázium (TÁG)* jelenlegi épülete 1996-ra készült el. A modern építmény hatalmas (kb. 17 000 m^2) mérete miatt rengeteg érdekes mérnöki megoldást alkalmaz. Takarékossági okokból (illetve részben a méretek miatt is) pl. nem építették ki a hagyományos csengőrendszert sem. Ehelyett kezdettől fogva az iskolarádióban szól az órák elejét és végét jelző szignál. De sajnos az idő eljárt a régi rendszer felett.

Több ok is indokolttá tette a lassan 18 éve működő berendezés leváltását:

1. A régi program egy mára igencsak **elavult** PC-n fut, ezáltal **nagy** az **energiafogyasztása** is.
2. Mivel a számítógépen a *Windows 98* fut ezért már évek óta **nincs hozzá frissítés** sem.
3. **A hardver kora** is eléggé aggasztó... Mi lenne ha egy nap már arra mennénk be, hogy a csengő hirtelen megadta magát?!
4. Sajnos **a csengőprogram funkciói sem tartottak lépést a korral**. Idén az iskolaigazgató elrendelte, hogy a 0. órára is be kell csengetni. Sajnos a szoftver erre nincs felkészítve, ezért a 7. órára gyakorlatilag nem csengetett mostanában, mivel csak 7 óra adatait tárolta (a 0. órával együtt pedig már 8 van). Ráadásul az automata kitöltési funkciója is használhatatlanná vált emiatt, ezzel okozva sok hosszú percnyi számolást és pötyögést oktatástechnikusunknak.

Ráadásul **arra sem emlékezett senki sem, hogy a régi rendszert ki programozta**, a forráskód pedig valószínűleg már réges-régen egy lemezformázás áldozatául esett. Ezzel meg is bélyegződött az előző csengő sorsa: már nem érdemes felújítani, helyette egy teljesen új rendszert kellett létrehoznunk.

Vegyük sorba, milyen feltételeket kell teljesítenie az általunk tervezett rendszernek!

A legelső szempont az **energia- és költséghatékonyság** volt. Már senki sem akart egy komplett PC-t használni arra a célra, amire gyakorlatilag egy sokkal egyszerűbb eszköz is képes. Ez az egyszerűbb eszköz pedig a ***Raspberry Pi*** (lásd 1. ábra). 3,5 wattos teljesítményével kb. 30 000 Ft-os vételára (mindennel együtt) 80 napon belül meg is térül, ezért tökéletes választásnak tűnt. Mivel egy monitor csak növelné a rendszer fogyasztását, illetve az adatbevitelt sem lehet igazán kényelmesen megoldani ezen a miniszámítógépen, ezért felmerült az ötlet, hogy **webes vezérlőfelület**et hozzunk létre. (Ez azonban felveti az **online biztonság** kérdését is...) Továbbá lényeges elem volt az is, hogy a már **meglévő rendszerhez** (erősítők, hangfalak, stb.) **illeszthető** legyen. Mindemellett **oldja meg a régi csengő problémáit**: kezelje a 0. órát, s legyen használható automatikus kitöltési lehetősége is. Továbbá **legyen** viszonylag **egyszerűen újraírható** a rendszer, ha megint módosítanának a csengetési renden.

6. ábra A Raspberry Pi

2. Megoldások

2.1. Szoftveres oldal

Ahhoz, hogy teljesíteni tudjuk azt a feltételt, hogy a programrendszer rugalmasan újraprogramozható legyen, elvetettük az általunk hagyományosan használt programnyelveket, és szkriptekben írtuk meg a szoftvert. Mivel a *Raspberry Pi*-n egy *Linux* disztribúció teljesít szolgálatot,

ezért a parancsértelmezője támogatja a *Bash* szkriptek használatát. Így ezen a nyelven íródott meg a csengetőprogram. Ez a programrész biztosítja azt, hogy megadott időpontokban megszólaljon az órákat jelző szignál. Ehhez bekapcsolja az erősítőket, majd lejátssza a megfelelő zenét. A szükséges pillanatok ideje egy szöveges fájlban tárolódik. Ennek az az oka, hogy szükség esetén kézzel is létre lehessen hozni egy konfigurációs fájlt.

Ennél azonban sokkal kényelmesebb, ha valaki egy gyönyörű, grafikus felületen viheti be a megfelelő adatokat (lásd 2. ábra). Telepítettünk hát egy *Apache* webszervert *Rapsberrynkre*, és *php*-ban írtunk néhány űrlapot és megfelelő feldolgozó szkriptet. Ezek a felhasználó által megadott adatokból egyetlen gombnyomásra megadják a csengetések időpontját, elkészítik a konfigurációs fájlt és el is indítják a csengető rutint. Amiatt, hogy a konfigurációs felületet hálózaton keresztül lehet elérni, biztosítani kell azt is, hogy „illetéktelenek" ne férjenek hozzá az iskola csengőjéhez. Ehhez a htaccess nevű eszközt használtuk, hogy jelszavas titkosítást alkalmazzunk a programrendszerünkre. Ez nem tökéletes védelem, de jól működik, és valószínűleg a diákok többsége nem ért (és nem is fog érteni) a feltöréséhez, szóval a mi célunknak pontosan megfelel.

7. ábra Online konfigurációs felület

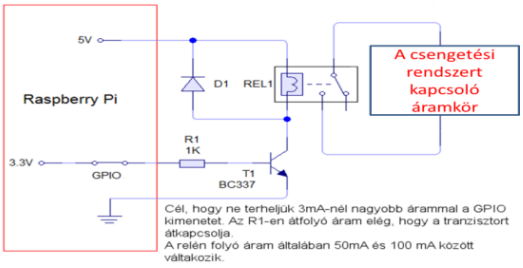
2.2. Hardveres oldal

A *Raspberry Pi*-nket alkalmassá kellett tennünk arra is, hogy az iskola már meglévő hangrendszerével kompatibilis legyen. A rendszernek be kell kapcsolnia 5 darab erősítőt mielőtt lejátszaná a csengő hangjait. Ezt a miniszámítógép egyik *GPIO* portjára kötött áramkörrel végeztetjük el. Olyan áramkört alkottunk (lásd 3. ábra), amely nem csak megvalósítja ezt a feladatot, hanem biztonságos is, hiszen a *Raspberry* NYÁK-lapját nem arra tervezték, hogy hatalmas áramerősségekkel küzdjön meg. Ezért egy kis áramú vezérlőkör végzi az erősítőket felkapcsoló, egyébként sokkal nagyobb árammal dolgozó második áramkör elindítását. Így a csengőrendszer üzembiztosan kapcsolódik az iskolarádió hálózatához.

3. Összefoglalás

Azt hisszük elmondhatjuk, hogy egy nagyszerű új rendszert sikerült létrehoznunk. Reméljük hosszú távú működése során is jól fog működni, és legalább annyi ideig fogja ellátni funkcióját, mint elődje. Hogy elkerülhessük az esetleges problémákat részletes dokumentációt is készítettünk az egész projektről. Ezekben a rendszergazda, az oktatástechnikus és tanáraink, esetleg utódaink, a diákok is részletes leírást találnak majd évekkel ezután is, hogy pontosan mit hogy kell beállítani, hogy teljesen jól működjön a program, ha esetleg át kellene írni vagy újra kellene telepíteni. Néhány szót szólnánk még a projekt nevéről is. A „*tAGBell*" cím kettős jelentésű. Egyfelől utal az iskola nevére (*TÁG*) és a csengőre (angol „*bell*" szó). Illetve a telefon feltalálójára is megemlékeztünk (***A**lexander **G**raham **Bell***) a névadással.

Szeretnénk megköszönni a segítséget *Hotzi Tibor, Kelemen Mihály* és *Bordi István* tanár uraknak, *Kovács István* oktatástechnikusnak és *Pelei Gergő*nek.



8. ábra Kapcsolási rajz

Nemzeti Fejlesztési Ügynökség
www.ujszechenyiterv.gov.hu
06 40 638 638

MAGYARORSZÁG MEGÚJUL

A projekt az Európai Unió támogatásával, az Európai Szociális Alap társfinanszírozásával valósul meg.