

Járműdinamikai mérő- és információs rendszer fejlesztése

Szauter Ferenc*, Székely János Ádám, Horváth Ernő

* Széchenyi István Egyetem,
H9026 Győr, Egyetem tér 1. (Tel.: +36 30 391-53-01, e-mail: szauter@gmail.com)

Abstract: Járműdinamikai mérő, adatgyűjtő eszközök kifejlesztése, az elektromos hajtású járművek igényei szerint. A rendszer komplex adat sorok rögzítésével, kiértékelésével pontosabb képet ad az alternatív járművek menetstabilitásáról és annak vizsgálatáról.

1. BEVEZETÉS

A járműdinamikai mérő- és információs rendszer kifejlesztésének egyik célja, két olyan eszköz létrehozása, ami belső-égésű motorral ellátott és alternatív hajtású járművek vizsgálatára is használható. Fejlesztéseink során sikerült két olyan eszközt, prototípust elkészíteni, ami teljesíti a feltételeket. Ez a két rendszer a Mobile Dynamic Measurement röviden (Mob DyMes) és a CANBUS Dynamic Measurement nevet kapta.

2. MOBILE DYNAMIC MEASUREMENT

2.1 Alkalmazási terület

A Mobile Dynamic Measurement főleg robogókhoz, illetve olyan járművekhez alkalmazható, ami nem rendelkezik CAN hálózattal.

A rendszert már az alternatív, elektromos meghajtású járművek vizsgálati és kísérleti igényei szerint fejlesztjük, ezért a mért jelek közül kiemelten fontosak a hajtás villamos jellemzői (feszültség, áram, teljesítmény), illetve az akkumulátor mindenkori állapotjellemzői, a BMS (Battery Management System) figyelése is.

2.2 Felépítés

Az eszközben GPS modul, RTC modul, MicroSD kártyafogó, háromdimenziós gyorsulás mérő szenzor, hőmérséklet szenzor, árammérő és feszültségmérő IC-k kaptak helyet.

A kijelző és beviteli eszköz egy OLED érintőkijelzővel valósult meg. Kijelzésre került az akkumulátorok és BMS rendszer állapota, hőmérséklet, aktuális sebesség, megtett út GPS koordináták, idő, a jármű dőlésszöge és ebből számított kanyarodási ív (ez a funkció egy nyomvonalon haladó járművek mérésénél használatos). A menü parancsok bevitelére „gombok” is helyet kaptak a kijelzőn. A hangfigyelmeztetés piezós hangszórón keresztül történik.

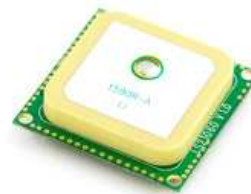


1. ábra: Érintőkijelzőn megjelenített adatok.

A berendezés egyedileg programozott operációs rendszerrel működik. Az adatfeldolgozást egy mikrovezérlőn keresztül történik. A mért jelek kondicionálása az eszközön belül valósul meg. Az így kinyert adat már a tényleges decimális paramétereket képviselnek, nincs szükség dekódoló programokra.

2.2.1. GPS

A berendezés „lelke” a GPS, ezért egy 66 csatornás 10Hz-es egység került beépítésre. Mivel az általunk felépített program a GPS adatokból számítja ki a sebességet és a megtett utat, így igazán pontos egységre volt szükségünk. A pontosítást különböző pozicionáló algoritmusok is segítik.



2. ábra A 66 csatornás GPS modul.

2.2.2. RTC modul

Napjainkban számos alkalmazásnak van szüksége dátum- és időinformációra. Az RTC modul, vagyis a Real Time Clock modul az idő és dátum paramétereket képes tárolni. Természetesen ezeket ki lehet nyerni a GPS adatokból is, de az a GPS adatfolyamát, kommunikációs sávját szűkítené.

Az RTCben egy DS1307 naptáráramkör és egy lítium akkumulátor található, ami 17 évig nem merül le.



3. ábra: Real Time Clock module

2.2.3. Háromdimenziós gyorsulás érzékelő

Az általunk beépítésre kerülő analóg háromtengelyes gyorsulás érzékelő szenzor +/- 3 g tartományban képes mérni. A felhasználási szándékainknak ez megfelelő intervallumot ölel át. mintavételezési rátája 1600Hz-ig növelhető.



4. ábra: Háromdimenziós gyorsulás érzékelő.

2.3. Programozása

A gyorsulás-jeladó mikrokontrollerjeként az Arduino Uno típust választottuk, ennek programozása Arduino IDE szoftverrel történik. Az Arduino IDE beépített C/C++ osztálykönyvtárral rendelkezik, ez a „Wiring”, amibe számos gyakori I/O művelet van előre kódolva. Maga a programozás C/C++ nyelven valósult meg. Egy megkötés van a programmal szempontjából: tartalmaznia kell két függvényt. Ezek pedig a következők:

- setup() - ami a program végrehajtása elején lefutó inicializáló rész
- loop() - ami pedig iteratíván futó rész

A programozás során a megszokott C szintaktika alapján használhatunk vezérlőket (if, else, for, while, stb), aritmetikai műveleteket (+, -, % stb), adattípusokat (void, byte, int, float, stb). Kiegészítve az IDE beépített függvényeivel, amelyek jók digitális és analóg I/O műveletekre, időzítéssel kapcsolatos műveletekre (mils(), delayMicroseconds(), stb), matematikai műveletekre (sin(), cos(), stb), valamint megszakításokra.

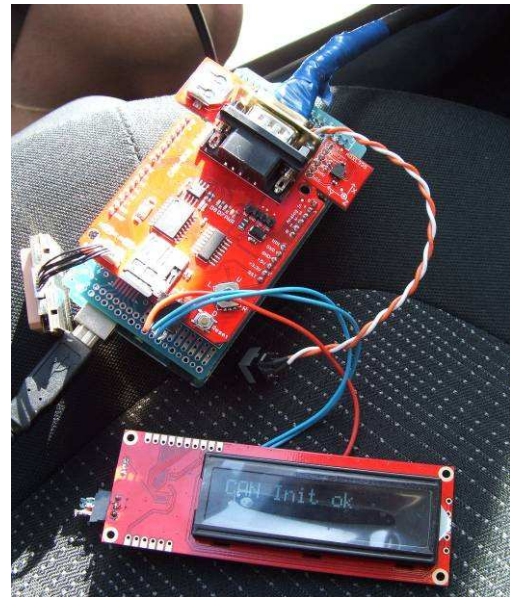
A programozás tanulására viszonylag jól dokumentált, bőséges szakirodalom áll rendelkezésre, valamint az interneten található sok végfelhasználó által fejlesztett program forrás-kódja is.

3. CANBUS DYNAMIC MEASUREMENT

3.1. Rendszer felépítése

A második kifejlesztett eszköz képes CAN-BUS (Controller Area Network BUS) rendszerrel való kommunikációra is, ami teljesen járműplatform független. Továbbá a rendszer GPS koordinátákat és háromdimenziós gyorsulás adatokat is rögzít időbélyegzővel ellátva egy memóriakártyára, úgy mint a Mob DyMes-nél. Ezáltal az eszközzel meghatározható a jármű egy adott pillanatban, a GPS koordinátái, motoradatai és gyorsulásértékből kiszámított járműdinamikai paraméterei. A rendszer lehetővé teszi, hogy járműdinamikai szimulációs programokban pontosan rekonstruálhatjuk a jármű mozgását. Egyik alkalmazási területként a villamos hajtásra átépítendő, de jelenleg belsőégésű motorral ellátott járművek pontos menetdinamikai szimulációjának későbbi összehasonlítása. Továbbá alkalmazható alternatív hajtású járművek menetparamétereinek meghatározására, vizsgálatára.

A megjelenítő eszköz egy 2soros 16 karakteres LCD kijelző, a beviteli eszköz egy joystick, aminek 4 iránya és nyomógomb funkciója van középállásban. A joystick-kel könnyedén navigálhatunk a rendszer menüjében. Bal- és jobb irányba való mozgatáskor a menü opciói között lépegethetünk. Középállásban gombnyomásra belép a kiválasztott menüre, lefele mozgatásra kilép belőle.



5. ábra: A prototípus, CAN hálózatra kötve egy személyautóban.

A rendszer változó mintavételezési sebességgel tudja gyűjteni az adatokat. Ez a mintavételezési ráta függ a rögzítendő pa-

paraméterek mennyiségétől. Maximális adatrögzítés esetén 0,5 másodpercenként, egy paraméter esetén 0,1 másodperces rögzítésre képes.

3.1.1. GPS

A CANBUS Dynamic Measurement Globális helymeghatározó rendszere, egy 20 csatornás 10hz-es GPS modul. Pontossága 1 méter.



6. ábra: 20 csatornás GPS modul

AZ RTC és a háromdimenziós gyorsulásmérő modulok meg-
egyeznek a Mob DyMes-ben használtakkal.

3.2. Program indítása és menüje

A program indítása után az üdvözlőképernyő jelenik meg. Ezután a joysticket balra vagy jobbra mozgatva léphetünk a menüben. Hat menü közül választhatunk.

Ezek a következők:

- SD-re rögzítés: Kártyára való rögzítés be- vagy ki- kapcsolása között választhatunk.
- SD beállítások: Kiválaszthatjuk, milyen paramétereket szeretnénk, illetve milyen formátumú fájlként rögzítse a rendszer a memóriakártyára.
- SD tartalom: A fájlokat és tartalmukat lehet megtekinteni.
- Gyorsulás monitor: az aktuális háromdimenziós gyorsulás értékeket, és a dőlésszögeket olvashatjuk le.
- CAN monitor: CANBUS-ra csatlakoztatás esetén a motoradatokat felügyelésre van lehetőség.
- GPS monitor: Pontos GPS koordináták megtekintése.
- Idő monitor: Idő lekérdezésére szolgál

3.3 Kimeneti fájl

Az eszközön az „SD beállítások” menüpont alatt kiválaszthatóak a rögzíteni kívánt paramétereket, így csak azok kerülnek a fájlba, amik számunkra fontosak. A kimeneti fájlt MicroSD memóriakártyára menti a rendszer táblázat formában. A fájl strukturálisan a következő oszlopokból áll:

- Időbélyeg
- Motorfordulatszám
- Motorhőmérséklet
- Járműsebesség
- MAF szenzor
- Lamda szonda feszültség értéke
- Nyomaték
- X tengely gyorsulásértéke
- Y tengely gyorsulás értéke
- Z tengely gyorsulás értéke
- GPS koordináták (LAT, LONG)

Ezen túl még programozható az eszközbe más paraméter is. A mentett fájl TXT, XML és CSV formátumú lehet.

3.4 A forráskód ismertetése

#define LABVIEW;

A szintén általunk készített LabVIEW program számára értelmezhető üzeneteket ad ki az eszköz soros portján. Amennyiben az eszköz SD kártyára rögzít adatokat és nincs szükség az élő feldolgozásra a jobb erőforrás kihasználás érdekében célszerű a definíciót "kikapcsolni" és ekkor az #ifdef LABVIEW és #endif közötti programrészek nem futnak le.

menuMegjelenites(); Ez a függvény a program futási logikájának része. Feladata a menü kijelzése.

void sdRogzit(); Ez a függvény SD kártyára menti a meghatározott adatokat, a megadott néven.

void keTorles(); Ez a függvény a képernyő (LCD) tartalmát törli le.

void poz(byte s, byte o); Ez a függvény a kurzort az s-sel jelzett sorba és az o-val jelzett oszlopba helyezi. Ezután az LCD-re írni kívánt

void beallit(byte s, byte o); Ezen függvény meghívásakor beállításokat végezhetünk, arra vonatkozóan, hogy milyen adatokat kerüljenek mentésre az SD kártyára.

void sdOlvas(); A függvény az SD kártyán található fájlból olvas ki. Soros portra kiíratható a teljes tartalom, azonban LCD képernyőre csak korlátozott lehetőség van kiírni.

gyorsulasKijelzes(); Ez a függvény a gyorsulási értékeket jelzi ki.

canKijelzes(); Ez a függvény a CAN értékeket jelzi ki.

gpsKijelzes(); Ez a függvény a GPS koordinátákat jeleníti meg.

oraKijelzes(); Ez a függvény az időt határozza meg.

void setup(); Minden Arduino program kötelező eleme a függvény, ebbe az inicializáláshoz szükséges műveletek kerültek.

void loop(); A program futási logikája kerül ebbe a függvénybe.

5. TOVÁBBI CÉLOK

5.1 Adatfeldolgozó program

A rendszerek kifejlesztése közben már elkezdtek olyan LabVIEW programokat írni, ami segít az adatfeldolgozásban, szimulációkban. A programba megnyitott kimeneti fájl egy pillanat alatt kiértékelésre, ábrázolásra kerül, két- és háromdimenziós diagramokon.

A gyorsulás értékekből származtatott menetdinamikai adatok, illetve motoradatokhoz algoritmusok fejlesztésére van szükség.

6. ÖSSZEFOGLALÁS

Kutatócsoportunk olyan új járműdinamikai mérő- és információs rendszert fejleszt, mely járműplatform független. A megvalósuló mobil berendezések egyedileg programozott

operációsrendszerrel futnak, a kezelése érintőképernyőn való parancs és adatbevitellel történik. A különböző mérendő jelek kondicionálása egyrészt a berendezésen belül történik. A rendszert már az alternatív, elektromos meghajtású járművek vizsgálati és kísérleti igényei szerint fejlesztjük, ezért a mért jelek közül kiemelten fontosak a hajtás villamos jellemzői (feszültség, áram, teljesítmény), illetve az akkumulátor mindenkor állapotlellemzői, a BMS (Battery Management System) figyelése. A rendszert felkészítjük a CAN-BUS (Controller Area Network BUS) rendszerrel való csatlakoztatásra és a mért-vezérelt jellemzők szintjén a kommunikálásra. A Széchenyi István Egyetemen folyó alternatív járműfejlesztésnél a a járműhajtást NI gyártmányú CompactRIO real-time/FPGA technológiás valós idejű operációs rendszerrel működő egység szeretnénk megoldani, ezért fontos a mérő-információs rendszer ez irányú kommunikációs kapcsolatának kiépítése is.

FELHASZNÁLT IRODALOM

[1] <http://arduino.cc>