

DEVELOPMENT OF A VEHICLE-DYNAMIC MEASUREMENT AND INFORMATION SYSTEM

F. SZAUTER[✉], J. Á. SZÉKELY, E. HORVÁTH

Széchenyi István University, Department of Automotive and Railway Engineering
H9026 Győr, Egyetem square 1, HUNGARY
[✉]E-mail: szauter@gmail.com

The aim was to develop vehicle-dynamic measurement and data collection tools in case of electric-drive vehicles. The system gives more detail oriented view of the driving stability of the alternative vehicles via the record of complex data.

Keywords: development, measurement, devices, CAN-bus, automotive, unique system development, arduino, LabVIEW

Introduction

The first aim of the developing vehicle-dynamic measurement and data collection system is to produce two devices, one of these is capable to examine an internal combustion engine powered vehicles and the other is for examine alternative vehicles. During the development process we were able to produce two prototypes that meets these conditions. We call these two systems Mobile Dynamic Measurement (or short: Mob DyMes) and CANBUS Dynamic Measurement.

Mobile Dynamic Measurement

Application area

A Mobile Dynamic Measurement can be applied mainly for scooters and for vehicles which do not support CAN protocol. The system was developed according to test and inspection purposes of alternative and electric-powered vehicles, so the most important from the measured signals were the following: the driving electrical characteristics (voltage, current, power) and the battery current status characteristics of the BMS (Battery Management System) monitoring.

Structure

The device has GPS module RTC module, MicroSD card slot, three-dimensional acceleration sensor, temperature sensor and ICs to measure current and voltage. The display and input device is with an OLED touch screen realized. It can display the battery status and the BMS system, temperature, current speed, distance travelled

GPS coordinates, time, vehicle steering angle and the date of this form (this feature is a measurement of tracks of vehicles used). On the display there are menu "command-keys" displayed as well. There is also an audio warning through a piezo-speaker.



Figure 1: The touch screen

The unit works with an individually programmed operating system. The data-processing is done through a microprocessor. The measured signal conditioning takes place within the device. The resulting data represents the actual decimal parameters, so there is no need for decoding programs.

GPS

One of the main components of the system is a 66-channel 10 Hz GPS. Since the program we have built calculates the speed and distance according to the GPS data, we needed a really accurate unit. The results are more accurate with the help of different positioning algorithms calculated from the acceleration sensor.

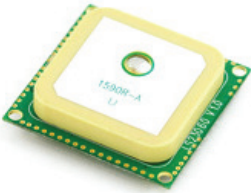


Figure 2: A 66-channel GPS module

RTC module

Today many applications need the date and time information. The RTC module (Real Time Clock module) can store time and date parameters. Of course, they can be obtained from the GPS data, but it would narrow down the GPS data stream communication. The DS1307 RTC has calendar-circuit and a lithium battery, which is not discharged for 17 years.



Figure 3: Real Time Clock module

Three-dimensional acceleration sensor

The built in three-axis analogue accelerometer sensor can measure acceleration in range +/- 3 g. As the matter of proper use of our intentions, this interval is enough and the sampling rate can be increased up to 1600 Hz.

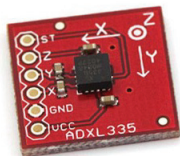


Figure 4: Three-dimensional acceleration sensor

Programming the system

The chosen microcontroller was Arduino Uno, which is programmable with the Arduino IDE environment. Arduino hardware is programmed using a Wiring-based language, similar to C++ with some simplifications and modifications, and a Processing-based IDE. The Arduino IDE comes with a C/C++ library called “Wiring”, which makes many common input/output operations much easier.

Arduino programs are written in C/C++, although users only need define two functions to make a runnable program:

- setup() – a function run once at the start of a program that can initialize settings
- loop() – a function called repeatedly until the board powers off

In the program we are free to use any normal control structures (if, else, for, while, etc.), arithmetic operators (+, -, % etc.), data types (void, byte, integer, float, etc.), not to mention the IDE’s build in functions, which allows us to perform digital and analogue I/O operations, timing operations (milis(), delayMicroseconds(), etc.), mathematical operations (sin(), cos(), etc.), and also interrupt operations. In case of learning the programming there is a relatively well documented literature available on the Internet and plenty source codes are also obtainable.

CANBUS Dynamic Measurement

The structure of the system

The second developed tool is capable to communicate with CAN-bus (Controller Area Network bus) system too, completely independent from vehicle platforms. It also records GPS coordinates and three-dimensional acceleration data with a time stamp to a memory card, like the Mob DyMes. This means vehicle at any given moment can be determined by the GPS coordinates, motor vehicle data and acceleration values calculated from dynamical parameters. The system allows us to accurately reconstruct the vehicle movement in vehicle dynamics software. One area of application can be the internal combustion engine vehicles which can be later electric driven redeveloped. The system could be used to compare the two states of these vehicles with precise driving dynamics simulation. Furthermore it can be applied for the determination and examination of alternative-powered vehicles parameters.

The display device is an LCD and has 2 rows, each row can display 16 characters, input device is a joystick, which has four directions and also has a button function in the central position. With the help of the joystick it is possible to navigate in the menu system. When left and right directions pressed we can move between menu options. We can select a menu by pressing the button and exit from a menu by moving down.

The system can only collect data with variable sampling rate. This sampling rate depends on the amount of parameters to collect. At maximum data recording speed it can record data in every 0.1 seconds, but if we want to collect every parameter it can only record in every 0.5 seconds.

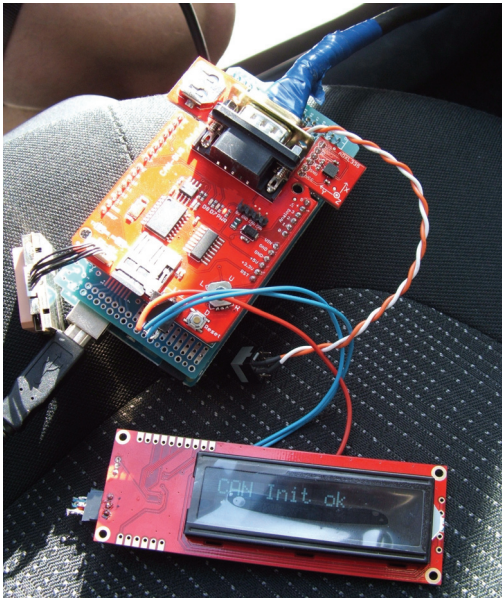


Figure 5: The prototype on CAN area network in a vehicle

GPS

The positioning system of CANBUS Dynamic Measurement is a 20-channel, 10 Hz GPS module. The accuracy is 1 meter. CANBUS Dynamic Measurement of Global Positioning System, a 20-channel, 10 Hz GPS module. Accuracy can be up to 1 meter.



Figure 6: 20 channel GPS module

The RTC and the three-dimensional accelerometer modules are the same as used in Mob DyMes.

The initialization and the menu of the program

After the initialization of the system it shows a welcome screen. If we move the joystick left or right we can select one of the menus. There are 7 menus to be selected. These are the followings:

- Record to SD card: Start/stop recording to the SD card
- SD settings: There can be selected which data to record and also the output format of the file
- SD content: A limited access to the file's content and also a directory listing
- Accelerator monitor: Gives us the possibility to see the current acceleration values or the angle
- CAN monitor: Gives us the possibility to see the current engine data provided by the vehicle's CAN system

GPS monitor: Gives us the possibility to see the current GPS values

Time monitor: Gives us the possibility to see the current time and date values

Output file

Entering tot the device's SD settings (*SD beállítások*) menu gives us the possibility to choose which data (parameters) to record into the output file. The system records the collected data to an output file on the Micro-SD card. The possible data (parameter) can be the following:

- Time stamp
- Engine Speed
- Engine temperature
- Vehicle Speed
- MAF sensor
- Lamda probe voltage
- Torque
- X-axis acceleration
- Y-axis acceleration
- Z axis acceleration
- GPS coordinates (LAT, LONG)

The short review on the source code

```
#define LABVIEW;
```

This definition is only necessary if the device is connected to a computer and there is a need to stream live data (such as acceleration, GPS coordinates etc.). It can be processed with our self-developed LabVIEW program. The type of the connection is basically a serial connection but it is realised in the USB port. If there is no need for live streaming the definition can be commented. In this case the commands between `#ifdef LABVIEW` and `#endif` won't be executed. It is also a possibility for better and faster debug of the program.

`menuMegjelenites();` This function displays the menu according to the joystick state.

`void sdRogzit();` This function saves all the data (parameters) which have been selected before.

`void keTorles();` This function provides a possibility to clean the LCD.

`void poz(byte s, byte o);` This function provides the possibility to put the cursor to the s row and o column. After that we can write to that point.

`void beallit();` This function provides a possibility to select which data can be stored on the SD card.

`void sdOlvas();` This function provides a limited access to the output file (because of the limitation of the 16x2 LCD screen)

`gyorsulasKijelzes();` This function provides live acceleration values. It can be displayed on the LCD screen.

`canKijelzes();` This function provides current engine data from the vehicle's CAN system.

`gpsKijelzes();` This function provides current GPS values. It can be displayed on the LCD screen.

oraKijelzes(); This function provides the current date/time values.

void setup(); This is a necessary function of all Arduino programs. It contains functions and operations that run once at the start of a program and who can initialize settings.

void loop(); This is also a necessary function of all Arduino programs. It contains functions and operations which are called repeatedly until the board powers off.

Further goals

During the development of system we already started to write LabVIEW programs, which help in data processing and simulation. One of the program is capable to analyse and display data from the output file the other can stream and display live data when the device is connected to the computer.

There is a need for further develop of the algorithm, to get practiser driving dynamics, and engine data.

The device has diagnostic functions also, and there is a possibility of further integration with AVL simulation environment.

Summary

Our researcher group is developing a new vehicle dynamics measurement and information systems which are vehicle platform-independent. These mobile systems have self-developed programs and have touch screen for controlling and displaying data. There are different signals to be measured which are conditioned within the unit.

The system was developed according to test and inspection purposes of alternative and electric-powered vehicles, so the most important from the measured signals were the following: the driving electrical characteristics (voltage, current, power) and the battery current status characteristics of the BMS (Battery Management System) monitoring. The system is prepared to the connection to the CAN-bus (Controller Area Network bus) system and also to measure these kinds of signals.

ACKNOWLEDGEMENTS

TAMOP-4.2.1/B-09/1/KONV-2010-0003: Mobility and Environment: Research in the fields of motor vehicle industry, energetics and environment in the Central- and Western-Transdanubian Regions of Hungary The Project is supported by the European Union and co-financed by the European Social Fund”

REFERENCES

1. A. R. DE SOUZA, A. C. PAIXAO, D. D. UZEDA, M. A. DIAS, S. DUARTE: The Arduino board: a low cost option for physics experiments assisted by PC (Revista Brasileira De Ensino De Fisica, 2011)
2. <http://arduino.cc/>
3. <http://en.wikipedia.org/wiki/Arduino>
4. A. THOMAS: Computer Hardware and Low Level Machine Language Programming, Integrated Graphic and Computer Modelling, 2008, 177–222
5. M. ZHOU, X. AO, J. WANG: Fault Diagnosis of Automobile Based on CAN Bus, Information and Automation, Communications in Computer and Information Science, 86, (2011), 317–323