International Journal of Engineering Research in Electronic and Communication Engineering (IJERECE) Vol 2, Issue 7, July 2015

Driver Assistance System Using Lab VIEW

[1]Jose J Edathala,[2]Megha Varghese,[3]Mereena Thomas,[4]Navya Athira Ram,[5]Vrinda Krishna
[1]Assistant Professor, Department of Electronics and Communication Engineering
[2] Department of Electronics and Communication Engineering
Amal Jyothi College of Engineering
Kanjirappally, India

Abstract: **This project aims in automising certain features of a car. The two features automised are the wiper system and headlamp system. Both of these are implemented using a graphical platform called LabVIEW.Windshield wipers play a key role in assuring the driver's safety during precipitation. The traditional wiper systems, however, requires driver's constant attention in adjusting the wiper speed and the intermittent wiper interval because the amount of precipitation on the windshield constantly varies according to time and vehicle's speed. The manual adjustment of the wiper distracts driver's attention, which may be a direct cause of traffic accidents. Similarly the automatic dimming of headlamp makes it effortless for the driver's thereby ensuring it to be a reliable mode of transport. Thus the project is an endeavor towards an effective design and development of an automatic windshield wiper system and headlamp system.**

Keywords—**Drivers assistance, LabVIEW, DAQ**

I. INTRODUCTION

Nowadays, safety in road traffic is an important topic for publicity as well as for vehicle manufacturers. Therefore, one main challenge is to develop Driver Assistance Systems (DASs) to support the driver in various conditions, e.g Automatic headlight control (AHC) for night time driving conditions and Automatic Wiper Control(AWC) actuated upon variation in refractive index of windshield glass.Driver assistance systems support overstrained and affected drivers and become more and more essential for series-production vehicles.

Thanks to the wake of the electronic and information technology evolutions, vehicles are expected to increase their capabilities of interacting with drivers. The development in automated driving endows vehicles to relieve drivers from undesired routines of driving task. A number of research programs have been aiming to develop various advanced technologies for driving-assistance systems in the world.

There are several advantages in providing driver assistance such as it 1)Reduces manual efforts 2)Reduces risk of accidents 3)Ensures safe and secure driving4)Makes driving more comfortable by reducing distraction5)Very reliable.

II. OVERLOOK ON LABVIEW

LabVIEW (short for Laboratory Virtual Instrument Engineering Workbench) is a system-design platform and development environment for a visual programming language from National Instruments.
The graphical language is named "G". Originally released for the Apple Macintosh in 1986, LabVIEW is commonly used for data acquisition, instrument control, and industrial automation on a variety of platforms including Microsoft Windows, various versions ofUNIX, Linux, and Mac OS X.

A. Dataflow Programming

The programming language used in LabVIEW, also referred to as G, is a dataflow programming language. Execution is determined by the structure of a graphical block diagram on which the programmer connects different function-nodes by drawing wires. These wires propagate variables and any node can execute as soon as all its input data become available. Since this might be the case for multiple nodes simultaneously, G is inherently capable of parallelexecution. Multi-processing and multi-threading hardware is automatically exploited by the built-in scheduler, whichmultiplexes multiple OS threads over the nodes ready for execution.

B. Graphical Programing

LabVIEW ties the creation of user interfaces (called front panels) into the development cycle. LabVIEW programs/subroutines are called virtual instruments (VIs). Each VI has three components: a block diagram, a front panel and a connector panel. The last is used to represent the VI in the block diagrams of other, calling VIs. The front

panel is built using controls and indicators. Controls are inputs – they allow a user to supply information to the VI. Indicators are outputs – they indicate, or display, the results based on the inputs given to the VI. The back panel, which is a block diagram, contains the graphical source code. All of the objects placed on the front panel will appear on the back panel as terminals. The back panel also contains structures and functions which perform operations on controls and supply data to indicators.Collectively controls, indicators, structures and functions will be referred to as nodes. Nodes are connected to one another using wires. The graphical approach also allows non-programmers to build programs by dragging and dropping virtual representations of lab equipment with which they are already familiar. The LabVIEW programming environmentmakes it simple to create small applications.The most advanced LabVIEW development systems offer the possibility of building stand-alone applications. Furthermore, it is possible to create distributed applications, which communicate by a client/server scheme, and are therefore easier to implement due to the inherently parallel nature of G.

Merits of LabVIEW

Interfacing to Devices

LabVIEW includes extensive support for interfacing to devices, instruments, cameras, and other devices. Users interface to hardware by either writing direct bus commands (USB, GPIB, Serial) or using high-level, device-specific, drivers that provide native LabVIEW function nodes for controlling the device.

ii)Code Compilation

In terms of performance, LabVIEW includes a compiler that produces native code for the CPU platform. The graphical code is translated into executable machine code by interpreting the syntax and by compilation. The LabVIEW syntax is strictly enforced during the editing process and compiled into the executable machine code when requested to run or upon saving. In the latter case, the executable and the source code are merged into a single file. The executable runs with the help of the LabVIEW run-time engine, which contains some precompiled code to perform common tasks that are defined by the G language. The run-time engine reduces compile time and also provides a consistent interface to various operating systems, graphic systems, hardware components, etc. The run-time environment makes the code portable across platforms.

iii)Large Libraries

Many libraries with a large number of functions for data acquisition, signal generation, mathematics, statistics, signal conditioning, analysis, etc., along with numerous graphical interface elements are provided in several LabVIEW package options. The number of advanced mathematic blocks for functions such as integration, filters, and other specialized capabilities usually associated with data capture from hardware sensors is immense. In addition, LabVIEW includes a text-based programming component called MathScript with additional functionality for signal processing, analysis and mathematics. MathScript can be integrated with graphical programming using "script nodes" and uses a syntax that is generally compatible with MATLAB.

iv)Parallel Programming

LabVIEW is an inherently concurrent language, so it is very easy to program multiple tasks that are performed in parallel by means of multithreading. This is, for instance, easily done by drawing two or more parallel while loops. This is a great benefit for test system automation, where it is common practice to run processes like test sequencing, data recording, and hardware interfacing in parallel.

III. DATA ACQUISITION

Data acquisition is the process of sampling signals that measure real world physical conditions and converting the resulting samples into digital numeric values that can be manipulated by a computer. Data acquisition systems (abbreviated with the acronym DAS or DAQ) typically convert analog waveforms into digital values for processing.The components of data acquisition systems include 1)Sensors that convert physical parameters to electrical signals.2)Signal conditioning circuitry to convert sensor signals into a form that can be converted to digital values.3)Analog-to-digital converters, which convert conditioned sensor signals to digital values.Data acquisition applications are controlled by software programs developed using various general purpose programming languages such as LabVIEW, BASIC, C, Fortran, Java, Lisp, Pascal. Stand-alone data acquisition systems are often called data loggers.There are also open-source software packages providing all the necessary tools to acquire data from different hardware equipment. These tools come from the scientific community where complex experiment requires fast, flexible and adaptable software.

NI my DAQ is a low cost portable data acquisition device that uses Ni LabVIEW based software instruments, allowing students to measure and analyze a real wordsignals. NI my DAQ is ideal for exploring electronics and taking sensor measurements. It provides analog inputs, analog outputs, digital input and output, audio, power supplies, digital multimeter functions in a compact USB device.

Figure 1: NI my DAQ

A. DAQ Hardware

DAQ hardware is what usually interfaces between the signal and a PC. It could be in the form of modules that can be connected to the computer's ports, cards connected to slots (S-100 bus, AppleBus, ISA, MCA, PCI, PCI-E, etc.) in the motherboard. Usually the space on the back of a PCI card is too small for all the connections needed, so an external breakout box is required. The cable between this box and the PC can be expensive due to the many wires, and the required shielding.

DAQ cards often contain multiple components (multiplexer, ADC, DAC, TTL-IO, high speed timers, RAM). These are accessible via a bus by a microcontroller, which can run small programs. A controller is more flexible than a hard wired logic, yet cheaper than a CPU so that it is permissible to block it with simple polling loops. For example: Waiting for a trigger, starting the ADC, looking up the time, waiting for the ADC to finish, move value to RAM, switch multiplexer, get TTL input, let DAC proceed with voltage ramp.

B. DAQ Device Drivers

DAQ device drivers are needed in order for the DAQ hardware to work with a PC. The device driver performs low-level register writes and reads on the hardware, while exposing API for developing user applications in a variety of programming environments.

IV. EXPERIMENTAL SETUP

The implementation of the automatic headlight control (AHC) and automatic wiper control (AWC) systems are done in LabVIEW.

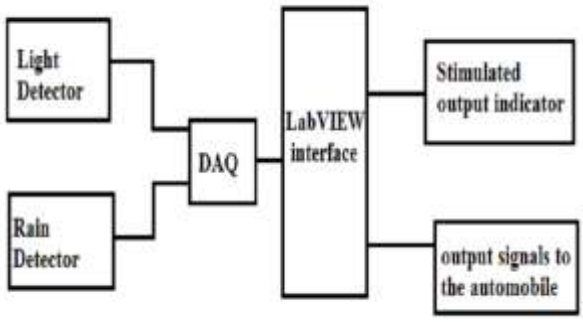

Figure 2. Flow of Sequence

A.Automatic Headlight Control (AHC)

The primary component used in AHC is the Light Dependent Resistor (LDR), whose resistance alters depending on the variation in light intensity.When the intensity of light falling on the LDR increases, its resistance decreases thus turning ON the transistor and providing minimal light output. However, when the intensity of light falling on the LDR decreases, its resistance increases, turning OFF the transistor and hence providing considerable light output. A potentiometer is used to vary the sensitivity of the LDR.The working of an AHC is illustrated in Figure 3.

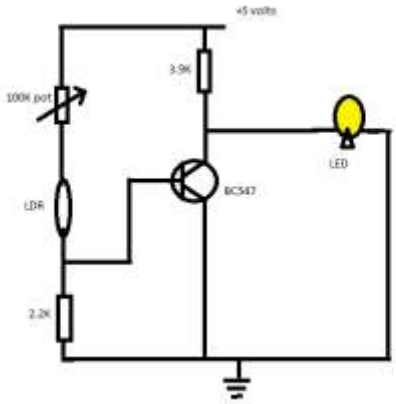

Figure 3.Automatic Headlight Control

B.Automatic Wiper Control(AWC)

A555 timer in the a-stable mode is the preliminary component used here.A 470K potentiometer is used alongside. Based on the variation in resistance given by the potentiometer there will be an output of variable duty cycle obtained. The working of an AWC is illustrated in Figure 4.

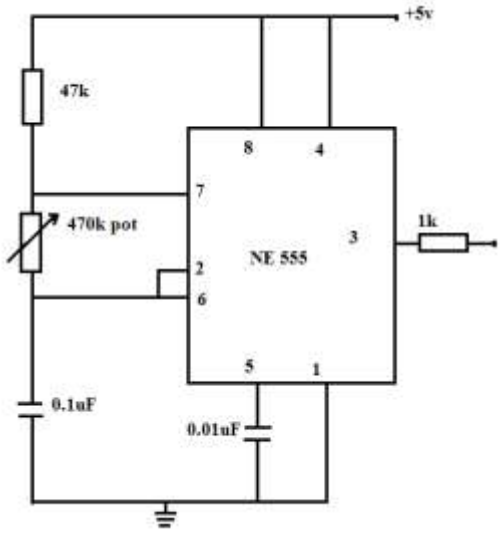

Figure 4. Automatic Wiper Control

V. LABVIEW IMPLEMENTATION

The output of both AHC and AWC are driven into LabVIEW using the NI my DAQ. It acquires the analog output signals of AHC in the form of voltage and the digital output of AWC in the counter format. The corresponding LabVIEW simulations are shown in Figure5-7.

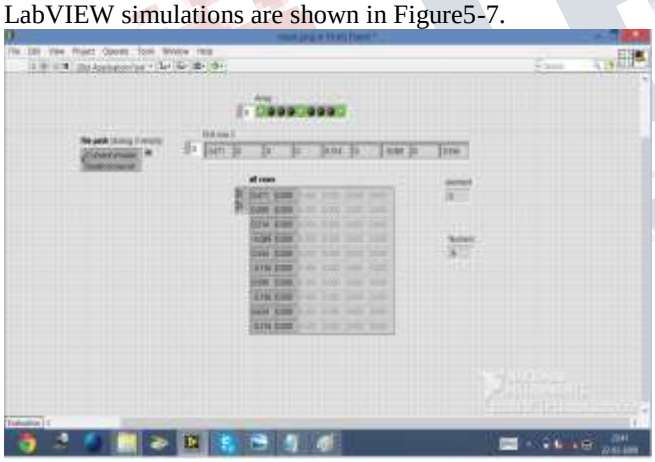

Figure 5: Front Panel

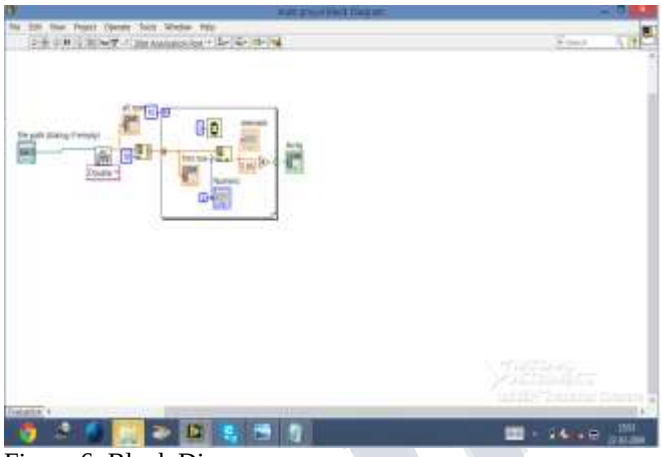

Figure 6: Block Diagram

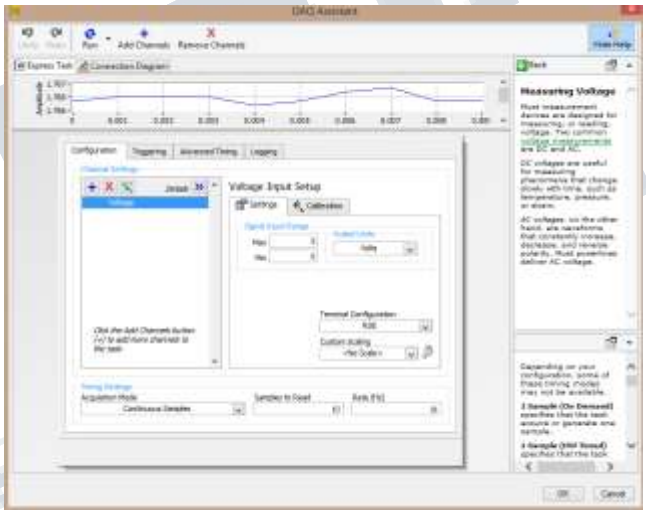

Figure 7:DAQ Assistant of AHC

CONCLUSION

The objective of the proposed work is to automize certain features of a car and thus facilitate a mechanism that assists the automobile drivers. This assistancemay prove to be an effective and reliable one. The software platform chosen for its implementation is LabVIEW. The advantage of this project is that it reduces the risks of accidents and loss of lives.

REFERENCES

[1] Ashik K.P and A.N. Basavaraju, 2014, " Automatic Wiperswith mist control", American Journal of Engineering Research,Volume-03, Issue-04,pp-24-34

[2]Mrs.Bhavna Ambudkar,2009, "Sensored Car", Second International Conference on Computer and Electrical Engineering

[3] Mr.Anil G Bansode, Prof S.O.Rajankar and Dr M.G.Ghatule, 2012, "Design and Development of smart automatic Windshield Wipers system :Fuzzy Logic Approach" , Research Inventy: International Journal of Engineering and Science ISSN: 2278-4721, Vol. 1, Issue 1, PP 14-20

● ● ●