COPS™ Based Automobile Instrument Cluster

National Semiconductor
Application Note 453
Venkata T. Gobburu
September 1986

ABSTRACT

Dedicated microprocessor systems find increasing applications in automobile instrumentation. Fuel injection systems, digital radio tuners and similar applications employing the microcontroller have become common place. This paper describes a cost effective microcontroller implementation of an automobile instrument cluster by the COPS group of National Semiconductor, Santa Clara. The instrument cluster provides a vacuum fluorescent display of the vehicle speed, engine RPM, odometers, battery voltage, engine oil pressure and the fuel level. A modular design involving a single microcontroller in conjunction with peripherals to aid in data acquisition from the transducers allows the quantities to be computed with high accuracies and displayed on a real time basis. The single microcontroller environment places severe restrictions on the availability of RAM and ROM. Coupled with the requirement of real time operation the application poses a non trivial challenge. A nonvolatile RAM accumulates the mileage covered. Hamming code techniques ensure the integrity of the data contained in the nonvolatile memory. Inclusion of diagnostics allows a rapid and thorough check against improper operation of the microcontroller, peripherals and the nonvolatile memory. This paper describes the implementation with a COP444L containing 128 nybbles of RAM and 2K bytes of ROM. A display updation rate of 16 Hz can be comfortably realized.

Over the microcomputer usage has diversified dramatically in its scope and breadth. Dedicated microprocessor systems find increasing application in automobile instrumentation and control. From its inception the automobile has acquired considerable sophistication. Increasing demands have been made of the car. Fuel efficiency, higher acceleration rates, simplicity of control and improved ride quality rank high in the demands made of the car. In response the automobile engine has evolved into a complex machine. Crude methods to control or monitor its performance no longer suffice. Microprocessor based fuel injection techniques and ignition control are becoming quite ubiquitous.

The automobile instrument cluster monitors the engine and regularly updates a status display for the operator's benefit. Pertinent information includes the vehicle speed, the engine crankshaft rotational speed, oil pressure in the engine cylinders, condition of the battery and the mileage accumulated. The instrument cluster provides a visual feedback link to the operator allowing corrective action to be initiated as the need arises.

THE AUTOMOBILE INSTRUMENT CLUSTER

The heart of the Automobile Instrument Cluster (AIC) lies in obtaining raw data from various transducers and manipulating it to a form suitable for feedback to the human operator. The feedback, normally visual, conveys the vehicle speed, the engine rpm, the engine temperature, oil pressure, the battery voltage and the odometer values. The AIC can be viewed as a collection of either inherently independent or weakly linked subtasks. Each subtask can be further partitioned into three blocks viz. of raw data collection, processing and displaying it. The component subtasks, in spite of their high degree of independence, can be grouped on the basis of signal available from the transducers. Grouping the subtasks modularizes the design. Partitioning the design in this manner highlights two groups, the first requires a frequency to be measured and the second a voltage level. The two major groupings are briefly examined.

Transducers for the vehicle speed monitor the driveshaft rotation. Computing the engine rpm involves measuring the crankshaft revolution rate. The two independent problems can be seen to basically consist of measuring revolution rates. Transducers based on Hall effect phenomena have been used with commendable success. Alternately the fact that mounting magnets around the driveshaft circumference generates a known number of pulses per shaft rotation can be used effectively. A normally open cam operated reed switch with closure to ground creates a simple revolution transducer. In all the cases the transducer generates a frequency proportional to the quantity under consideration. Obviously some signal conditioning is required before using the frequency with digital components. The describing function can be simply stated as

$$V = k \times f \qquad (1)$$

where

V is the quantity under measurement, the vehicle speed or the engine rotational speed
k is a proportionality constant
f is the transducer freqeuncy output

The proportionality constant, k, can be suitably modified to include changes back and forth between British and metric units.

The problem of measuring the transducer output frequency can be restated to be one of measuring the time period. In case of digital frequencies the equation (1) can be rewritten as

$$V = k/(Ton + Toff) \qquad (2)$$

where

Ton is the ON time and
Toff is the OFF time

while the remaining symbols retain their definition from the earlier equation.

The remaining quantities such as the engine temperature, oil pressure, battery voltage and available fuel prove to be slow changing ones. The lower dynamics allow them to be transduced as voltage level signals. Equation (3) states the underlying relation and closely resembles the equations stated above.

$$P = k \times v \qquad (3)$$

where

v is the voltage output of the transducer
P is the quantity under measurement
k is the proportionality constant

Evaluating the accumulating mileage depends indirectly upon the vehicle speed subtask. Integrating the signal from the vehicle speed transducer over time allows the mileage to be accumulated. The associated problems of storing the odometer information and ensuring its integrity require error correcting techniques. They are covered in a later section of the paper.

COPS™ and MICROWIRE™ are trademarks of National Semiconductor Corp.

RRD-B30M105/Printed in U. S. A.

SYSTEM DESCRIPTION

The COPS Group of National Semiconductor, Santa Clara, offers a wide array of microcontrollers and peripherals to suit this application. Judicious selection of peripherals to aid the microcontroller can reinforce the partitioning suggested earlier to considerably simplify the implementation. *Figure 1* presents a functional block diagram of the AIC.

A COP444L four bit microcontroller provides the necessary computing and decision making capability. Equipped with 128 nybbles of RAM space organized in a matrix fashion and 2K ROM space for storage of the control program, the COP444L operating at an instruction cycle rate of 16 microseconds sequentially obtains information from the peripherals and formats the manipulated results to be manageable by the display drivers. Transducers for the vehicle speed and the engine speed provide proportional frequency signals. Two COP452 peripherals, placed in a Waveform Measure Mode, track the ON time and OFF time of the conditioned transducer outputs. Voltage level signals available from the transducers for the engine temperature, oil pressure, battery condition and the fuel tank can be monitored by a COP438, an eight channel A/D converter. An electronically erasable non volatile RAM, the COP494, allows the odometer information to be stored safely under power down conditions.

A combination of LEDs, vacuum fluorescent displays and high intensity lamps comprise the optical elements of the AIC Standard eight segment alphanumeric and bargraph format displays have been used. A 32 segment LED bargraph, controlled by a MM5450 static display driver, displays the engine rpm. Eight segment alphanumeric vacuum fluorescent displays are used for the vehicle speed and the odometer values. Sixteen segment vacuum fluorescent bargraph displays are used for the engine temperature and available fuel quantity. The battery voltage and oil pressure utilize eight segment vacuum fluorescent bargraph displays. Any potentially dangerous situations detected by the COP444L are underlined by high intensity lamps. Five COP470 display drivers multiplex the various displays under the microcontroller's orchestration.

Single pole single throw switches allow the user to select between the British or the metric units, the trip or the accumulated odometer and reset the trip odometer.

SYSTEM DIAGNOSTICS

Diagnostics aid in isolating faulty components within a system. The algorithmic nature of the diagnostic procedure allows it to be implemented via a microprocessor. A great deal of attention has been focused on diagnostics as considerable cost savings can accrue from a microprocessor based scheme minimizing human involvement. Programming the AIC, in addition to its normal functions, with self test capabilities increases its potential for high volume applications. Normally diagnostics imply using independent means to evaluate the system's performance. Attempting to incorporate self test capabilities necessitates adopting an ‘‘inside out’’ strategy. A basic kernel is first evaluated as functioning correctly. Over iterations the kernel expands by establishing correct operation of other modules.

The AIC implementation described in this paper has an extensive repertoire of diagnostics to check the microcontroller and ensure correct operation of the peripherals. The probability of the microcontroller ROM failing proves to be negligibly small compared to a fault developing in the hardware interconnections. Also the idea of encoding in ROM the algorithm to check ROM data proves suspect. Control program stored in the ROM forms the kernel assumed to be functioning correctly. Writing and reading back an alternating pattern of ones and zeros in the microcontroller RAM checks for leakage of data into adjacent locations. Applying a known voltage, derived locally, to one of the four unused channels on the A/D converter allows it to be tested. The architecture of the COP452 peripherals consists of two independent register-counter pairs. The counters count down from the initial value. To test the COP452 both the register counter pairs have to be checked. By placing the two in a Duty Cycle Mode, the counters can be loaded with initial values from the registers and set to count down. The contents of the counters after a predetermined delay can detect incorrect operation of the device. A fault at the level of a register-counter pair can thus be isolated.

The COP494 stores the odometer information. It becomes vital to maintain the integrity of the information stored in the nonvolatile memory. Continuous use of particular locations in the COP494 can result in failures, typically bit dropouts. It is imperative to be capable of recovering from such errors. Requiring a single COP494 unit to last at least the expected lifetime of the vehicle influences the design of the storage scheme. The AIC implementation described in this paper depends upon Hamming encoding techniques to provide single bit error recovery. Subsequent to recovering from a single bit error all data transactions are carried out from a new location. A flashing display sequence alerts the operator of the occurrence of a non-recoverable error. Suspending all normal functions during such conditions can be used to force the vehicle to be taken to an authorized dealer. Breaking up the odometer data into sections allows updating of particular sections as opposed to restoring the whole every time. Such a strategy maximizes the lifetime of the nonvolatile memory.

SOFTWARE DESCRIPTION

The functional objectives of the AIC and the hardware required to realize them have been detailed in earlier sections of the paper. A summary of the software features completes the description and aids in developing a global understanding of the AIC. The AIC software, written in COP microcontroller assembly language, reflects the modular nature of the problem. The finite amount of memory of ROM space available on the COP444L coupled with real time operation requirements makes programming the AIC a non-trivial problem. Each subtask grouping has been organized as a distinct block of code. The microcontroller sequentially processes each subtask. A brief examination of the salient features follows.

It must be borne in mind that the COP452 peripheral captures an instantaneous picture of the frequency. The strength of the magnets, mounted circumferentially on the driveshaft to transduce revolution rate, cannot be precisely controlled. As a result the transducer, although generating a fixed number of pulses per revolution of the driveshaft, produces a pulse train showing both pulse period and duty cycle variations. Directly using the pulse period from the

COP452 leads to erroneous values of the vehicle speed. The computed vehicle speed, under steady vehicle speed conditions, shows excursions on either side of the nominal value. The first AIC implementation studied the application of an essentially single pole filter with different damping constants to exclude the oscillations. Although a sufficiently damped filter can effectively reduce the oscillations the scheme was discarded in lieu of the resulting degradation in response time. The solution lies in basing the vehicle speed computation on pulse period measurements averaged over consecutive pulses. Since the number of pulses per revolution is known, eight in this case, averaging the pulse period over this number minimizes the steady state error and responds fast. The nature of the solution affects the software organization. It falls upon the microcontroller to sample the conditioned output of the transducer and obtain pulse periods for eight consecutive pulses. To achieve this the software adopts a foreground-background organization. Monitoring the transducer output to catch the consecutive pulses forms the background job. The normal functions of the AIC form the foreground job. Additionally a minimal sampling rate has to be maintained to ensure that even at highest attainable vehicle speeds the microcontroller measures consecutive pulses.

The AIC electronically stores the odometer information in the non-volatile memory. Loss of odometer integrity can be disastrous. Consequently the ability to recover from errors in the non-volatile memory becomes very important. The AIC depends on single bit error correcting Hamming coding methods to avoid loss of information. The algorithm processes the odometer nybble fashion and simplifies the related problems of encoding the data prior to storing it and decoding the composite for data retrieval to trivial table lookups. LQID, a powerful member of the microcontroller instruction set, allows an eight bit value to be looked up based on the key value in the addresed RAM location. To minimize ROM space both the encoding and the decoding sections of the algorithm share the same error table and code for table lookups.

The remaining sections of the AIC software, also exhibit a block structure, do not prove to be as subtle. The straight forward code includes routines such as multiplications and divisions to help in the computations and routines allowing the microcontroller to communicate serially over the MICROWIRE™ with the peripherals.

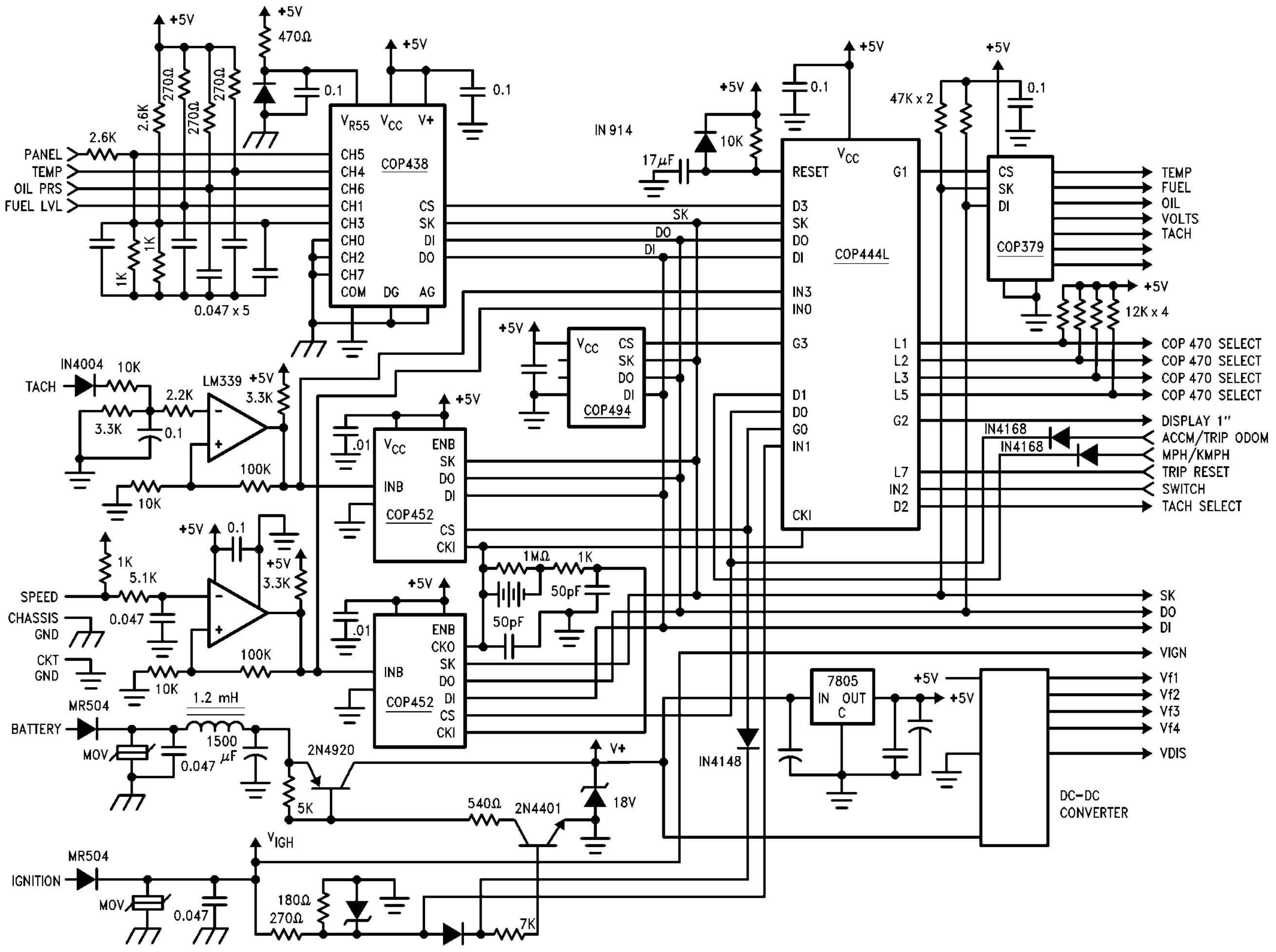
RESULTS AND CONCLUSIONS

The AIC implemented via the COP444L approximately uses 2K of ROM space. The COP444L, running at an instruction cycle time of 16 microseconds, sequences through all the functions in 228 milliseconds. The resulting display updation rate of approximately 4 Hz can be trivially increased to 16 Hz by replacing the COP444L with the equivalently packaged COP440. Table I presents in tabular form the accuracies and speeds at which the different measurements are done. It also shows the proportional speed increases obtainable.

The minimal number of peripherals used combined with the inclusion of diagnostics and error correction emphasize its low cost capabilities. The results serve to validate the feasibility of a cost effective microcontroller based Automobile Instrument Cluster.

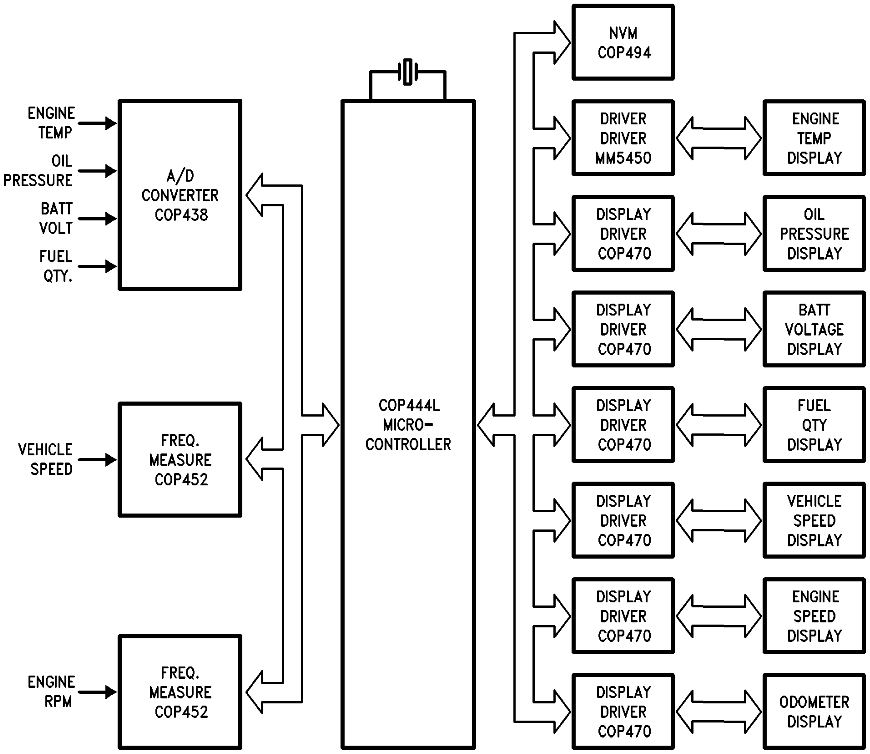
TABLE I. Comparison of Speed and Resolution of Measurements Taken with the COP444L and the COP440

	Measurements with a COP444L		Measurements with a COP440	
	Time Taken μsecs	Resolution Bits	Time Taken μsecs	Resolution Bits
1. Engine rpm	768	17	192	17
2. Vehicle Speed	768	17	192	17
3. Engine Temperature	256	8	64	8
4. Oil Pressure	256	8	64	8
5. Battery Voltage	256	8	64	8
6. Fuel Quantity	256	8	64	8



TL/DD/8798–1

FIGURE 1. Functional Block Diagrams of the AIC



TL/DD/8798–2

FIGURE 2

LIFE SUPPORT POLICY

NATIONAL'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF THE PRESIDENT OF NATIONAL SEMICONDUCTOR CORPORATION. As used herein:

1. Life support devices or systems are devices or systems which, (a) are intended for surgical implant into the body, or (b) support or sustain life, and whose failure to perform, when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury to the user.
2. A critical component is any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

National Semiconductor
Corporation
1111 West Bardin Road
Arlington, TX 76017
Tel: 1(800) 272-9959
Fax: 1(800) 737-7018

National Semiconductor
Europe
Fax: (+49) 0-180-530 85 86
Email: cnjwge@tevm2.nsc.com
Deutsch Tel: (+49) 0-180-530 85 85
English Tel: (+49) 0-180-532 78 32
Français Tel: (+49) 0-180-532 93 58
Italiano Tel: (+49) 0-180-534 16 80

National Semiconductor
Hong Kong Ltd.
13th Floor, Straight Block,
Ocean Centre, 5 Canton Rd.
Tsimshatsui, Kowloon
Hong Kong
Tel: (852) 2737-1600
Fax: (852) 2736-9960

National Semiconductor
Japan Ltd.
Tel: 81-043-299-2309
Fax: 81-043-299-2408

National does not assume any responsibility for use of any circuitry described, no circuit patent licenses are implied and National reserves the right at any time without notice to change said circuitry and specifications.