

The Project54 Common Interface for the Intelligent Transportation Systems Data Bus

Michael E. Martin, Francis C. Hludik, W. Thomas Miller
University of New Hampshire, ECE Department
Kingsbury Hall, Durham, NH 03824

Abstract – Police cruisers have a variety of aftermarket equipment such as lights, sirens, radar units, radios, etc. Most police aftermarket equipment was not designed with integration in mind. In order to allow the integration of electronic equipment in police cruisers a common interface for connecting aftermarket car electronics to an Intelligent Transportation Systems Data Bus (IDB) was designed and implemented. The interface hardware uses Version 2.0B of the Control Area Network (CAN) protocol as the method of bi-directional data transmission on the IDB. As a means to connect the aftermarket devices to the IDB the interface has two options. The first type of connection offered is a serial port that uses standard RS-232 signaling. The second type of connection is a port that offers five parallel TTL level signals that can be used for a variety of applications including switching. The common interface is currently being installed in several NH State Police Cruisers. Initial tests show the system to be reliable.

INTRODUCTION

Project54 is a collaborative research and development effort between the University of New Hampshire and the New Hampshire Department of Safety and is supported by the U.S. Department of Justice. The project involves integrating embedded mobile computing equipment and wireless networking into New Hampshire State Police Cruisers. The system uses voice technology, as well as integration of car-based electronics systems to offer advanced support for NH State Troopers [1].

Police cruisers have a variety of aftermarket equipment such as lights, sirens, radar units, radios, etc. Most police aftermarket equipment was not designed with integration in mind. The messaging between electronic devices in the cruiser can be standardized by making sure that the messaging protocols between them are standardized, and so is the bus structure that is used to transmit messages. An example of such standardization is the use of the Intelligent

Transportation System Data Bus (IDB) standard. The IDB standard prescribes both the hardware components and the protocol needed to accomplish control tasks and to exchange information between the car's systems. For example locking the doors can be accomplished by sending out a particular command on the IDB bus from the lock button to a door-locking actuator.

The purpose of the effort presented in this paper was to create a common interface between aftermarket electronics using the IDB standard. Almost no aftermarket devices are able to connect to the IDB, however most of them have serial and/or parallel inputs and outputs. Therefore we developed an embedded device that can interface to electronic products in a serial and parallel fashion, and can communicate with the IDB. Using this device, messages from the IDB can be translated and sent to a connected device, and responses from that device can be put on the IDB, as shown in Figure 1.

BACKGROUND

The time to market for an automobile can be several years. Automobile manufacturers have to predict what consumers will want in their vehicles five or more years from the current time. The IDB standard developed by the

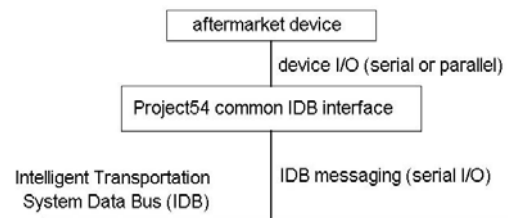


Figure 1 The common IDB interface unit allows IDB messages to be passed to the aftermarket device and responses from the device to be put on the IDB.

The authors wish to thank the US DOJ for funding Project 54 through grant #1999-DD-BX-0082.

Society of Automotive Engineers hopes to allow for the addition of aftermarket products to an automobile by the consumer at any point during the lifecycle of the vehicle [2]. The Control Area Network (CAN) was chosen as the data bus standard. CAN is widely used in many industries today including manufacturing. CAN has already made inroads into the automotive industry as a means of reducing vehicle weight by reduction of wires needed in the automobile's electrical system. The Volvo S60 currently uses CAN for control of its throttle-by-wire system [3]. An effort similar to Project54, ALERT at Texas A&M also used the IDB standard to integrate mobile electronics for first responders [4].

CONTROL AREA NETWORKS

The Project54 common IDB interface hardware uses Version 2.0B of the CAN protocol as the method of data transmission on the IDB. The CAN protocol, developed by Robert Bosch GmbH, uses differential signaling that is ideal in noisy environments such as an automobile. All nodes on a CAN bus are able to transmit and receive data. Multiple CAN nodes can request the bus simultaneously. Nodes with a lower transmitting identifier have a higher priority for control of the bus. A binary countdown method is used to solve contention between two devices

on the network trying to transmit. When a message is broadcast all nodes on the bus receive the message. Each node determines based on the identifier if the message will be accepted and be processed. Each CAN message can contain up to 8 bytes of user information. Longer messages can be transmitted by segmentation of the data. The maximum transmission rate is 1 Mbit/s with a bus length of 40 m. Lower transmission rates can be used to improve on the maximum transmission length. The transmission rate for the Project54 application will be 250 kbit/s as specified by the IDB standard [5].

DESIGN & DEVELOPMENT

Standard designs for CAN controllers are widely available. These designs were used to create a design specific for Project 54. Two issues that were necessary to consider in the initial designs for this controller were versatility and size. There is an enormous amount of after market equipment, many examples of which can be found in Figure 2 that needs to fit into a limited amount of space inside each police cruiser. That equipment ranges from aftermarket police applications such as the radar and radio to the embedded PC and associated peripherals like a LCD touchscreen and data storage. Each car could require up to a dozen Project54 common IDB interface units depending on the police cruiser's equipment configuration. Figure 3

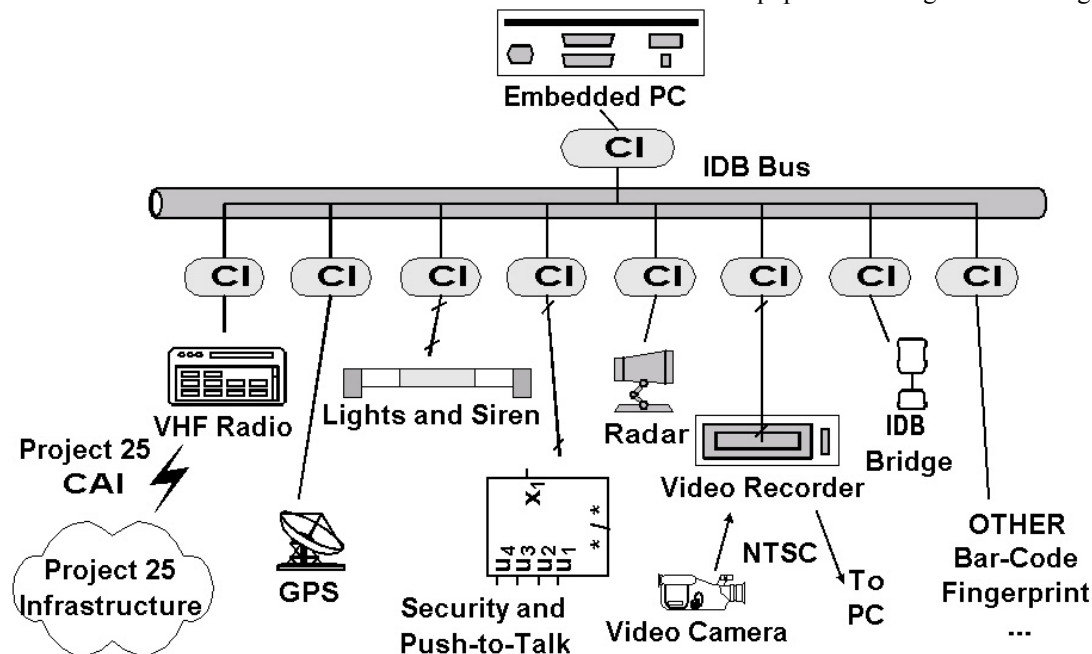


Figure 2 - Aftermarket devices in a police cruiser connected using the IDB. The individual devices are connected to the IDB using the common interface.

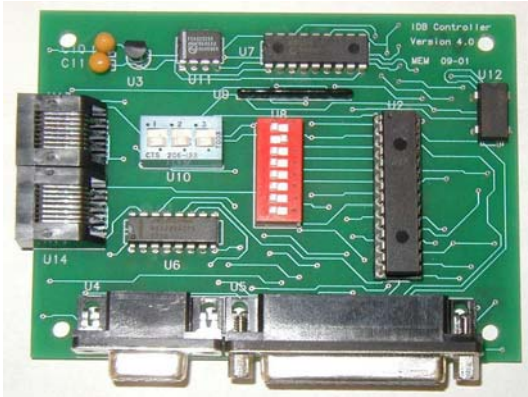


Figure 3 - Current common IDB interface board

shows the resulting design and some of the common IDB interface's important features. Initial designs of the common IDB interface used a single male DB9 connector to connect to the IDB as specified by the IDB standard. This bus configuration proved to be difficult to implement in the confined space of a police cruiser. Each common IDB interface now has two RJ45 jacks that are used to daisy chain each of the common IDB interfaces together to form the IDB as is shown in Figure 4. One aftermarket device would be connected to each common IDB interface shown in Figure 4. Standard category 5 cabling is used to make the connection between each of the common IDB interfaces on the IDB. This connection also supplies the power needed to run the common IDB interfaces. Standard category 5 cables have four twisted pairs. The differential signal and power / ground are each carried on two twisted pairs. This redundancy will allow the system to continue to operate in the case of partial cable failure. Each board draws about 45 mA of current from a 13.8 V supply [6].

As a means to connect the aftermarket devices to the IDB, the interface has multiple options and configurations. The multiple options and configurations allow the common IDB interface to connect to a wide variety of aftermarket devices. The first type of connection offered is a serial port that uses standard RS-232 signaling. The serial port can be configured to use hardware handshaking, if desired by means of an on-board single pole double throw (SPDT) switch. The common IDB interface can setup to support baud rates of 1200, 2400, 4800, 9600, 19,200 or 38400 bps. The second type of connection is a port that offers five parallel TTL level signals that can be used for a variety of applications. One such application is a switch for a relay driven lightbar. Power can also be

supplied to some aftermarket devices via this port at voltage levels of +5 V and +13.8 V. The current that can be sourced is limited. This port also offers a means of in-circuit programming and updating of the on-board microcontroller.

The flow of data between the aftermarket device and the IDB is handled by a microcontroller. Currently two versions of software are used for the Project54 common IDB interface's microcontroller. One version is used on-board the common IDB interface hardware connected to the embedded PC of the Project54 system as is shown in Figure 2. The Project 54 Application Manager is a Windows based application that runs on the embedded PC [7]. The Application Manager generates packets based on user input and system requirements. The Application Manager also interprets the data that is relayed serially to the embedded PC from the IDB through the embedded PC's common IDB interface. That data may range from a license plate query result to the current status of the emergency warning lights. In order to interpret the data the common IDB interface connected to the embedded PC sends additional information about the data packet to the embedded PC such as the data source and packet size. Also in order to transmit data on the IDB to its proper destination the embedded PC's common IDB interface receives packet data, packet destination and packet size information from the embedded PC.

All other Project54 common IDB interfaces on the IDB use a version of the microcontroller code that is designed to connect to the aftermarket devices. The IDB is designed to be invisible to all the aftermarket devices on the IDB. Simply stated the devices do not know that they are connected to the IDB. For example an aftermarket serial device such as the GPS would communicate with the common IDB interface in the same manner that it would if it was connected directly to a computer's serial port. This allows the devices to be connected to the IDB via the common IDB interface without alteration of the aftermarket devices.

During initialization the microcontroller examines the five address switches to set up the device's two unique IDB addresses. Consideration is being given to hardcoding the device's address into the microcontroller software to allow for simplification of the setup and to increase the number of parallel TTL level

signals to ten. This initialization process also sets up serial and/or parallel communication with the device. The first of the two addresses represents the serial data address. If the serial address of the Project54 common IDB interface is utilized the data contained within that packet will be sent out the serial port of the interface. The second of the two addresses is the command address of the Project54 common IDB interface. The command address uses the packet data to perform some action either by controlling internal states of the common IDB interface or controlling the aftermarket device that is attached to that particular Project54 common IDB interface. The serial port of the common IDB interface can be enabled or disabled to control the flow of data onto the IDB. A common IDB interface reset command is available to flush the common IDB interface data buffers. One of the most useful commands concerning the internal state of the common IDB interface is a ping command. When a common IDB interface is 'pinged' it sends out a packet that states that the device is active on the IDB and gives the version number of the microcontroller code that is being used. Controlling the aftermarket device requires application specific functions that are included in the microcontroller code that can perform the desired action according to the codes that are sent via the IDB. For example the common IDB interface connected to the lightbar may receive a Front Strobes On command from the Application Manager via the IDB. The common IDB interface would interpret the command to set the TTL level signals to a specific pattern that would cause the lightbar relay driver to turn the Front Strobes On [8].

TESTING

A variety of tests have been performed to ensure proper operation in the overall system implementation. Environmental testing has been carried out to guarantee IDB operation in the rigors of the New Hampshire environment. It was found that the IDB would operate at temperatures as low as 40° C from a cold start and that the IDB would continue to run at temperatures exceeding 120° C. The IDB was also subjected to large amounts of traffic generated by multiple devices on the network in order to make certain that no data is lost or corrupted during periods of high network usage. The common interface is currently being installed in several NH State Police Cruisers. Initial tests show the system to be robust and reliable in the actual operating environment.

CONCLUSION & FUTURE WORK

The Project54 common interface is used to connect aftermarket devices in the police cruiser to the IDB. The controller hardware was developed to incorporate a multitude of device types onto the IDB. Figure 2 gives a graphical representation of some of the devices that are being integrated into the system as part of Project 54. At the center of the system is an embedded PC. This embedded PC performs control of all devices on the IDB. The embedded PC uses the IDB to communicate with the devices. The aftermarket devices in a police cruiser range from standard serial devices, such as GPS, radar, etc., to relay driven light bars.

The long-term goal of Project54 is to

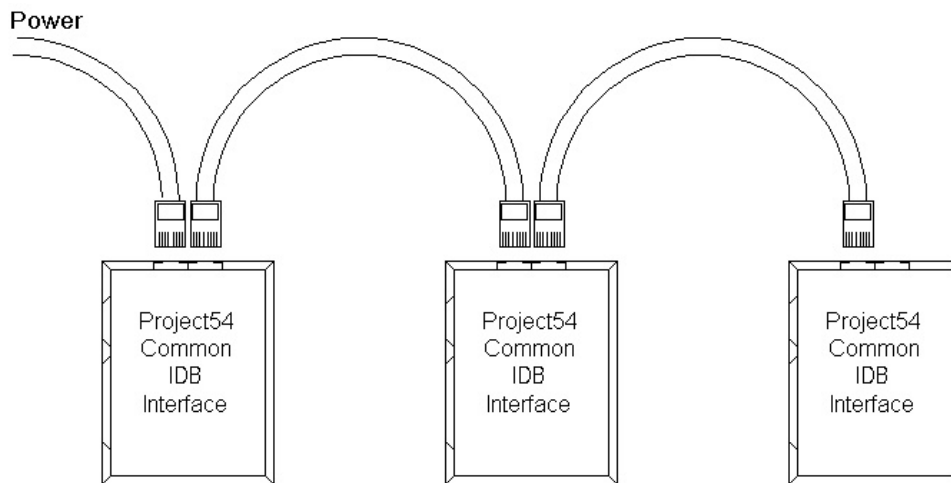


Figure 4 - IDB Configuration

implement this system in the entire NH State Police vehicle fleet. In order to maintain the system effectively a series of tools are being developed to compliment the IDB. Foremost among these tools will be a common IDB interface that can be connected to any existing IDB for purposes of troubleshooting a network. This particular common IDB interface will relay serially the IDB traffic including packet data and packet origin to a Windows based application. The application will allow the user to examine network traffic and to ping common IDB interfaces on the network for current operating states of that common IDB interface. Another tool that was discussed earlier is the updating of the common IDB interface's microcontroller through the one of the onboard ports. This allows the common IDB interface to be updated in its actual operating environment.

REFERENCES

- [1] UNH CATLab. Consolidated Advanced Technologies for Law Enforcement Program (CATLAB). 2000. 4 Feb. 2002. <<http://www.catlab.unh.edu/>>
- [2] "SAE J2355 ITS Data Bus (IDB) Architecture Reference Model Information Report," SAE International, May 1999
- [3] Grimshaw, Paul, "Savvy Engineering for the Socially Responsible," The Ottawa Citizen 26 Jan. 2001: Wheels Section.
- [4] ALERT. ALERT Homepage. 8 Feb. 2002. <<http://alert.tamu.edu/>>
- [5] Robert Bosch GmbH. CAN Homepage. 2002. 4 Feb. 2002. <<http://www.can.bosch.com/>>
- [6] Michael E. Martin, "The Project54 Common Interface for the Intelligent Transportation Data Bus Version 4," Technical report ECE.P54.2001.3, ECE Department, University of New Hampshire, 2001
- [7] A. L. Kun, W. T. Miller, W. H. Lenharth, "Project54: Introducing advanced technologies in the police cruiser," Proceedings of the 2002 IEEE Semiannual Vehicular Technology Conference (VTC Spring 2002), Birmingham, AL, May 6-10, 2002
- [8] Michael E. Martin, "Common IDB Interface Software Updates," Technical report ECE.P54.2002.1, ECE Department, University of New Hampshire, 2002