www.ijecs.in
International Journal Of Engineering And Computer Science ISSN:2319-7242
Volume 4 Issue 4 April 2015, Page No. 11215-11218

RTOS Based Fault Diagnosis And Efficient Data Transfer On Priority Basis Using Can

K.T Indhupriya B.E, G Balaji M.Tech
Ii Year, M.E-Embedded System Technologies,
Sri Shakthi Institute Of Engineering And Technology, Coimbatore.
Mouly.Indhu@Gmail.Com

Assistant Professor, Department Of Ece,
Sri Shakthi Institute Of Engineering And Technology, Coimbatore.
Bala.Bharatham@Gmail.Com

***Abstract*—CAN protocol is a broadcast communication protocol and it is used for automotive application. CANalyzer is a software tool which is used for analysis and stimulation of bus communication. It is used to check whether and what type of communication is occurring on the bus. It is used to send or log data. CANalyzer is a high cost tool and it is not suitable for small applications. This paper demonstrates the implementation of priority based message transfer using CAN protocol and also demonstrates the faulty messages in the system without using CANalyzer. The system uses µC/OS II RTOS and ARM.**

Keywords—CAN protocol, CANalyzer, µC/OS II, ARM, RTOS .

I. Introduction

The CAN protocol is based on a bus topology, and only two wires are needed for communication over a CAN bus. The bus has a multi master structure where each device on the bus can send or receive date [2]. Only one device can send data at a time while all others are listen. If two or more devices attempt to send data at the same time, the one with the highest priority is allowed to send its data while the others return to receive node.

CAN protocol supports fault detection and fault confinement features [1]. But architecture of CAN protocol does not support redundancy in channel or media.

In practice all the messages are stored in queue. So this paper implements priority scheduling algorithm.

II. Scheduling algorithm

PRIORITY SCHEDULING

Each process is assigned a priority. Process with highest priority is to be executed first and so on as shown in fig.2.1. Processes with same priority are executed on first come first serve basis. Priority can be decided based on memory requirements, time requirements or any other resource requirement.

Process	Arrival Time	Execute Time	Priority	Service Time
P0	0	5	1	0
P1	1	3	2	3
P2	2	8	1	8
P3	3	6	3	16

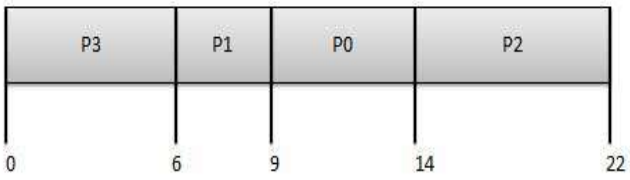

Fig 2.1 Priority table

III. Block diagram

BLOCK DIAGRAM

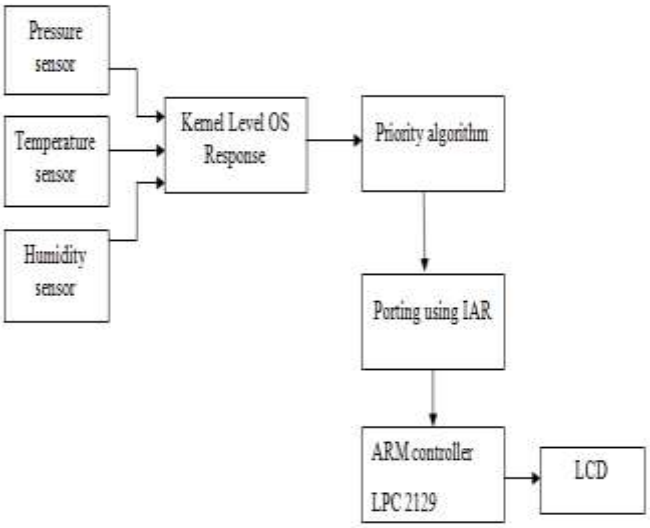

Fig 3.1 Block diagram

By using priority algorithm, data are transferred to the LCD through CAN bus and also to detect the fault in the sensor data. Each sensor called as a task as shown in fig.3.1.

PRESSURE SENSOR

A pressure sensor measures pressure, typically of gases or liquids. Pressure is an expression of the force required to stop a fluid from expanding, and is usually stated in terms of force per unit area. A pressure sensor usually acts as a transducer it generates a signal as a function of the pressure imposed.

US 300 is a low cost pressure sensor. Accuracy is 0.1% and operating temperature is -40°C to 105°C as shown in fig.3.2.

Fig 3.2 Pressure sensor

TEMPERATURE SENSOR

LM 35 is a low cost temperature sensor and it's output voltage is directionally proportional to the temperature. Temperature range is -55°C to 150°C and operating voltage is 4 V to 30 V as shown in fig.3.3.

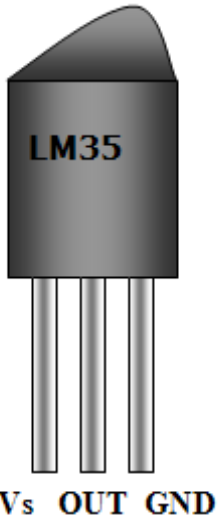

Fig 3.3 Temperature sensor

HUMIDITY SENSOR

The HIH-4030 humidity sensor is a great way to measure humidity. It draws very low current, and is simple to use. The output is a simple analog voltage, and when read with the ADC pin of a microcontroller, can be very easily translated into an accurate measure of relative humidity as shown in fig.3.4. The output voltage is directly proportional to the relative humidity.

Fig 3.4 Humidity sensor

µC/OS II

µC/OS II is a highly portable, ROMable, very scalable, preemptive real-time, deterministic, multitasking kernel. It can manage up to 64 tasks (56 user tasks available). It has connectivity with µC/GUI and µC/FS (GUI AND File systems for µC/OS II). It is ported to more than 100 microprocessors and microcontrollers. It is simple to use and simple to implement but very effective compared to the price/performance ratio. It supports all type of processors form 8-bit to 64-bit.

IAR EMBEDDED WORKBENCH:

IAR embedded workbench is a sophisticated and easy to use development tool for embedded application. It has IAR C/C++ compiler, assembler, linker, text editor, C-SPY debugger. With its built in ARM specific code optimizer, IAR embedded workbench generates very efficient and reliable code for ARM devices [3].

Core support for ARM Cortex-M0, Cortex-M0+, Cortex-M1, Cortex-M3, Cortex-M4(F), Cortex-R4(F), Cortex-R5, Cortex-R7, Cortex-A5, Cortex-A7, Cortex-A8, Cortex-A9, Cortex-A15, ARM7(E), ARM9(E), ARM11, Secure Core[3].

Device support including peripheral register definitions and flash loaders for over 2000 devices from analog devices, atmel, energy micro, free scale, Fujitsu, SiLabs, Infineon, Nuvoton, NXP, ON Semi, Samsung, ST, Texas Instruments, Toshiba, etc.

LPC 2129

LPC 2129 is a 32 bit ARM7 TDMI processor with real time emulation and embedded trace support with 256kb of flash memory. With their compact 64 pin package, low power consumption, various 32-bit timers, 4 channel 10-bit ADC, 2 advanced CAN channels, PWM channels and 46 GPIO lines with up to 9 external interrupt pins these microcontrollers are particularly suitable for automotive and industrial control applications as well as medical systems and fault-tolerant maintenance buses [4].

FEATURES

- 32-bit ARM7 TDMI-S microcontroller in a tiny LQFP64 package.
- 16kb on-chip static RAM.
- 256kb on-chip Flash program memory. 128-bit wide interface/accelerator enables high speed 60 MHz operation.
- In-system programming (ISP) and In-Application programming (IAP) via on-chip boot-loader software.
- Embedded ICE-RT interface enables breakpoints and watch points. Interrupt service routines can continue to execute while the foreground task is debugged with the on-chip Real Monitor software.
- Embedded Trace Macro cell enables non-intrusive high speed real-time tracing of instruction execution.
- Two interconnected CAN interfaces with advanced acceptance filters.
- Four channel 10-bit A/D converter with conversion time as low as 2.44 ms.
- Mutliple serial interfaces including two UARTs (16C550), Fast I2C (400kbits/s) and two SPIs.
- 60 MHz maximum CPU clock available from programmable on-chip Phase-Locked loop with settling time of 100 ms.
- Vectored Interrupt Controller with configurable priorities and vector addresses.
- Two 32-bit timers (with four capture and four compare channels), PWM unit (six outputs), real time clock and watchdog.
- Up to forty-six 5v tolerant general purpose I/O pins. Up to nine edge or level sensitive external interrupt pins available.
- On-chip crystal oscillator with an operating range of 1 MHz to 30 MHz.
- Two low power modes, Idle and Power-down.
- Processor wake-up from Power-down mode via external interrupt.
- Individual enable/disable of peripheral functions for power optimization.
- Dual power supply:
    1. CPU operating voltage range of 1.65 V to 1.95 V (1.8 V ±0.15 V).
    2. I/O power supply range of 3.0 V to 3.6 V (3.3 V ± 10 %) with 5 V tolerant I/O pads.

CAN CONTROLLER FEATURES

- Data rates up to 1 Mbit/s on each bus.
- 32-bit register and RAM access.
- Compatible with CAN specification 2.0B, ISO 11898-1.
- Global acceptance filter recognizes 11 and 29-bit Rx identifiers for all CAN buses.
- Acceptance filter can provide Full CAN-style automatic reception for selected standard identifiers.

LCD

In this project 16 x 2 LCD is used. It has 16 rows and 2 columns. It has two registers such as command and data register. The command register stores the command

instructions given to the LCD. A command is an instruction given to LCD to do a predefined task like initializing it, clearing its screen, setting the cursor position, controlling display, etc. The data register stores the data to be displayed on the LCD. The data is the ASCII value of the character to be displayed on the LCD.

It has 8 data lines (DB0-DB7) and 3 control lines (RS, R/W, E). If RS is high data register is selected if it is low means command register is selected. If R/W is high means read operations is performed if it is low means write operation is performed.

IV. Conclusion

Priority wise the sensor data will be sent through CAN bus. In case of any fault occurs in the sensor value means it will be displayed in the LCD. So the bus activity can be achieved by without using CANalyzer tool. So cost of the product can be reduced for small applications.

V. References

[1]. Supriya Kelkar, Raj Kamal, "Implementation of Data Reduction Technique in Adaptive Fault Diagnosis Algorithm for Controller Area Network", International conference on CSCITA, 2014.

[2]. Controller Area Network (CAN) Specification, Version 2.0, Robert Bosch, GmbH, Postfach, Stuttgart, 1991.

[3]. www.iar.com

[4]. S. Rajeswari,"Design and Implementation of Can Bus for Distributed Injection Molding Machine Based on Arm Controller", Middle-East Journal of Scientific Research, 2014.

[5]. Leung, J. Y. T., and Whitehead, J., "On The Complexity of Fixed-Priority Scheduling of Periodic Real-Time Tasks", Performance Evaluation **2**(4), pp.237-250 (December 1982).

[6]. Tindell, "Fixed priority Scheduling of Hard Real-Time Systems", YCST 94/03, DPhil Thesis, Department of Computer Science, University of York (1993).

[7]. Audsley, N.,Burns, A.,Richardson,M.,Tindell, K. and wellings, A.,"Applying New Scheduling Theory to Static Priority Pre-emptive Scheduling", Software Engineering Journal 8(5) pp. 284-292 (September 1993).