AUTOMOTIVE SAFETY SYSTEM

A PROJECT REPORT

Submitted by

VEERAMANI.N **Reg. No: 13BEC161**

WASHEEM AGRAM.P.A **Reg. No: 13BEC175**

SARAVANAKUMAR.T **Reg. No: 13BEC201**

SATHYASEELAN.K **Reg. No: 13BEC202**

in partial fulfilment for the award of the degree

Of

BACHELOR OF ENGINEERING

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

KUMARAGURU COLLEGE OF TECHNOLOGY

(An Autonomous Institution Affiliated to Anna University, Chennai)

COIMBATORE - 641 049

APRIL 2017

BONAFIDE CERTIFICATE

Certified that this project report "AUTOMOTIVE SAFETY SYSTEM " is the bonafide work of **VEERAMANI.N , WASHEEMAGRAM .P.A, SARAVANAKUMAR.T** and **SATHYASEELAN. K** who carried out the project work under my supervision. Certified further that to the best of my knowledge the work reported herein does not form part of any other project or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE

Mr. R. Navaneethakrishnan M.E,M.B.A.,

Assistance Professor/ECE,

Department of ECE ,

Kumaraguru College of Technology,

Coimbatore-641049.

SIGNATURE

Dr. K. Malarvizhi M.E,Ph.D.,

Head Of The Department,

Department of ECE,

Kumaraguru College of Technology

Coimbatore- 641049.

The candidates with Register No: 13BEC161, 13BEC175, 13BEC201, 13BEC202 are examined by us in the project viva-voce examination held on …….

INTERNAL EXAMINAR

EXTERNAL EXAMINAR

ACKNOWLEDGEMENT

We express our sincere thanks to the Management of Kumaraguru College of Technology and Joint Correspondent **Shri. Shankar Vanavarayar** for the kind support and for providing necessary facilities to carry out the project work.

We would like to express our sincere thanks to our beloved Principal **Dr. R. S. Kumar, Ph.D.,** Kumaraguru College of Technology, who encouraged us in each and every step of the project.

We would like to thank **Dr. K. Malarvizhi M.E., Ph.D.,** Head of the Department, Electronics and Communication Engineering, for her kind support and for providing necessary facilities to carry out the project work.

We wish to thank with everlasting gratitude to our project Co-ordinator **Dr.M.Ramalatha, M.S., Ph.D.,** Department of Electronics and Communication Engineering for her consistent support throughout the course of this project work.

We are greatly privileged to express our deep sense of gratitude and heartfelt thanks to our Project Guide **Mr. R. Navaneethakrishnan M.E., M.B.A.,** Department of Electronics and Communication Engineering for her expert counseling and guidance to make this project to a great deal of success and also we wish to convey our regards to all teaching and non-teaching staff of ECE Department for their help and cooperation.

Finally, we thank our parents and our family members for giving us the moral support and abundant blessings in all of our activities and our dear friends who helped us to endure our difficult times with their unfailing support and warm wishes.

ABSTRACT

Our project automotive safety systems is based on establishing communication between sensors and objects and the engine for the vehicle is interfaced with the arm7. Most of the people are using vehicles which have become the most important part in our life for comfort transportation . The system has two modules Master and Slave, they are communicating through Controller Area Network (CAN) protocol. And for the communication CAN (transmitter & reciever) which is controlled by ARM7 and sensor are interfaced with it. At once the signal is received by the sensors it indicates to the LCD display and buzzer.

TABLE OF CONTENTS

CHAPTER NO	TITLE	PAGE NO
	ABSTRACT	**iv**
	LIST OF FIGURES	**vii**
1	**Introduction**	**1**
2	**Block Diagram**	**2**
3	**Hardware Description**	**3**
	3.1 ARM7 Family	3
	3.1.1 ARM7 TDMI	4
	3.1.2 Instruction Pipeline	5
	3.1.2.1 Memory Interface	5
	3.1.2.2 Memory Format	5
	3.1.2.3 Performance Code density and operating sates	6
	3.1.3 Operating Modes	6
	3.1.3.1 Coprocessor	6
	3.1.3.2 Debug Features	7
	3.1.4 ARM7 TDMI Processor core	7
	3.1.4.1 Registers	7

3.1.4.2 Modes and exception 7

3.1.4.3 Processor States 8

3.1.4.4 Exceptions 11

3.1.4.5 Status Register 12

3.1.5 LPC2148 microcontroller 12

3.1.5.1 LPC2148 Microcontroller Architecture 13

3.1.5.2 Pin diagram 14

3.1.5.3 Architecture Overview 15

3.2 Control Area Network

3.2.1 Need for CAN 16

3.2.2 CAN Message Frame 17

3.2.3 Priority 18

3.3 Sensors

3.3.1 Ultrasonic Sensor 18

3.3.2 Temperature Sensor 19

3.3.2.1 Features 20

3.3.3 Fuel Level Sensor 21

3.3.4 Pressure Sensor 22

	3.4 LCD Module	23
4	**Software Tools**	
	4.1 Keil c	27
	4.2 Flash Magic	28
5	**Photograph Of Working Model**	**29**
6	**Working principle**	**30**
7	**Conclusion**	**31**
	Reference	**32**

LIST OF FIGURES

CHAPTER NO	TITLE	PAGE NO
2.1	Block diagram of automotive safety system	2
3.1	ARM7TDMI Core Diagram	4
3.2	Register Organization in ARM State	9
3.3	Register Organization in Thumb State	10
3.4	LPC2148 Micro Controller Architecture	13
3.5	LPC2148 Micro Controller Pin Diagram	14
3.6	Structure of CAN	17
3.7	Message Frame	17
3.8	Ultrasonic Sensor	19
3.9	Temperature Sensor	20
3.10	Fuel level sensor	21
3.11	Pressure Sensor	22
3.12	LCD Display	24
3.13	Circuit Diagram	25
5.1	Photograph of Working Model	29

CHAPTER 1

INTRODUCTION

With rapidly changing computer and information technology and much of the technology finding way into vehicles. They are undergoing dramatic changes in their capabilities and how they interact with the drivers. Although some vehicles have provisions for deciding to either generate warnings for the human driver or controlling the vehicle autonomously, they usually must make these decisions in real time with only incomplete information. So, it is important that human drivers still have some control over the vehicle. Advanced in-vehicle information systems provide vehicles with different types and levels of intelligence to assist the driver.

The driving is make easier and safety and reduce the human efforts .To make travelling is easy and safe. We see every day thousands of road accidents in this accidents as many as thousands of peoples injured in a world's More than hundreds of people die and many people are disabled for live life normally this is a result of lack of speed control and violating the road rules. Although vehicles have provisions warning and alert the driver using buzzer this for taking right decision or controlling the vehicle autonomously, they usually must make these decisions in real time with only complete information. This is Important that human drivers control over the vehicle and check the parameters in vehicle on screen at the same time of driving, parameters like temperature, fuel.

CAN protocol (bus) are used for data transmission, it is Application domain is from high speed network to low cost multiplex wiring because of its special capabilities.

CHAPTER 2

Block diagram

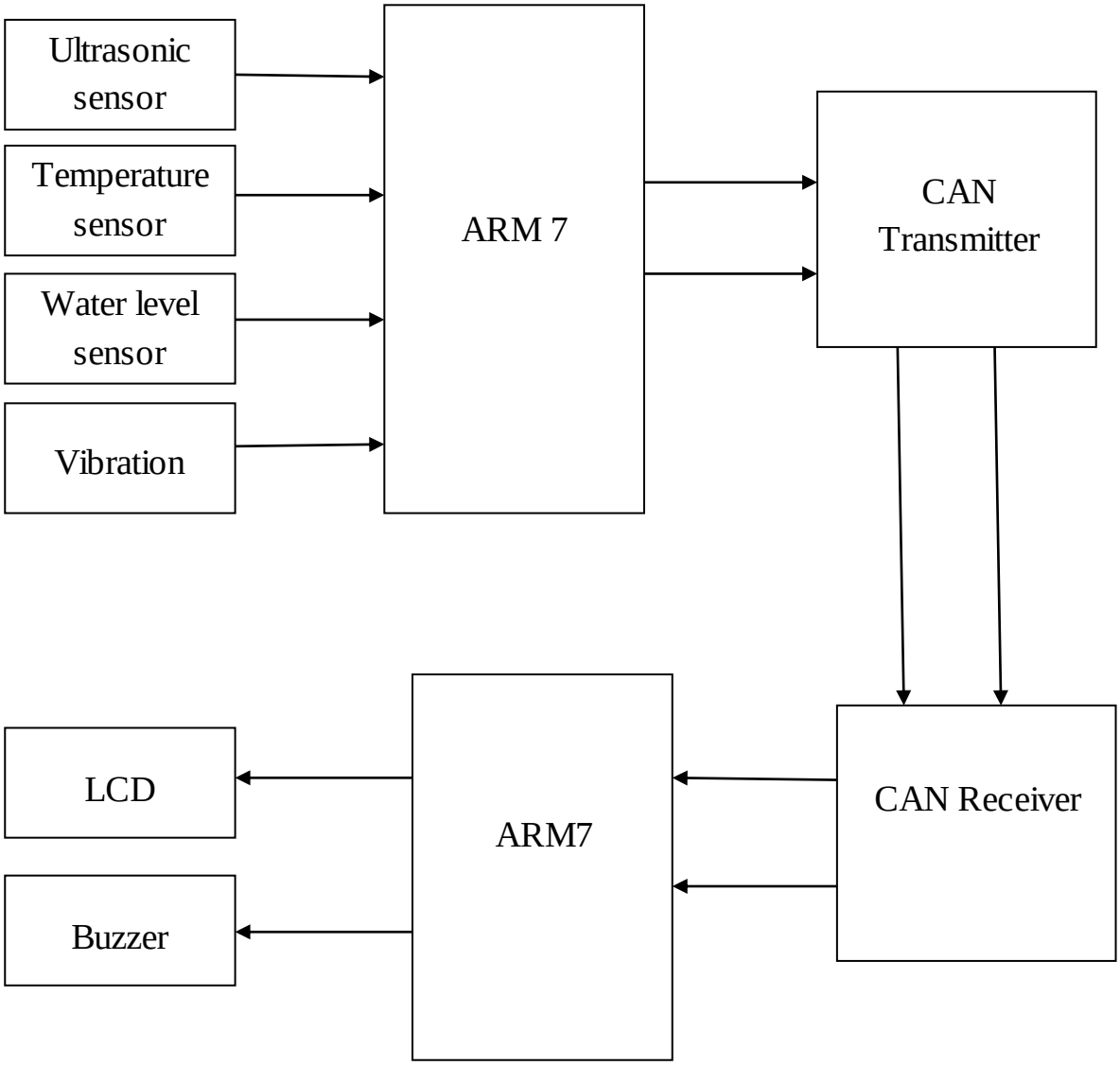


Fig 2.1 block diagram of automotive safety system

CHAPTER 3

Hardware Description

3.1 Arm7 Family

The ARM7 family includes the ARM7TDMI, ARM7TDMI-S, ARM720T, and ARM7EJ-S processors. The ARM7TDMI core is the industry's most widely used 32-bit embedded RISC microprocessor solution. Optimized for cost and power-sensitive applications, the ARM7TDMI solution provides the low power consumption, small size, and high performance needed in portable, embedded applications.

The ARM7TDMI-S core is the synthesizable version of the ARM7TDMI core, available in both VERILOG and VHDL, ready for compilation into processes supported by in-house or commercially available synthesis libraries. Optimized for flexibility and featuring an identical feature set to the hard macro cell, it improves time-to-market by reducing development time while allowing for increased design flexibility, and enabling >>98% fault coverage. The ARM720T hard macro cell contains the ARM7TDMI core, 8kb unified cache, and a Memory Management Unit (MMU) that allows the use of protected execution spaces and virtual memory. This macro cell is compatible with leading operating systems including Windows CE, Linux, palm OS, and SYMBIAN OS.

The ARM7EJ-S processor is a synthesizable core that provides all the benefits of the ARM7TDMI – low power consumption, small size, and the

thumb instruction set – while also incorporating ARM's latest DSP extensions and Jazelle technology, enabling acceleration of java-based applications. Compatible with the ARM9™, ARM9E™, and ARM10™ families, and Strong-Arm® architecture software written for the ARM7TDMI processor is 100% binary-compatible with other members of the ARM7 family and forwards-compatible with the ARM9, ARM9E, and ARM10 families, as well as products in Intel's Strong ARM and x-scale architectures. This gives designers a choice of software-compatible processors with strong price-performance points.

3.1.1 ARM7 TDMI

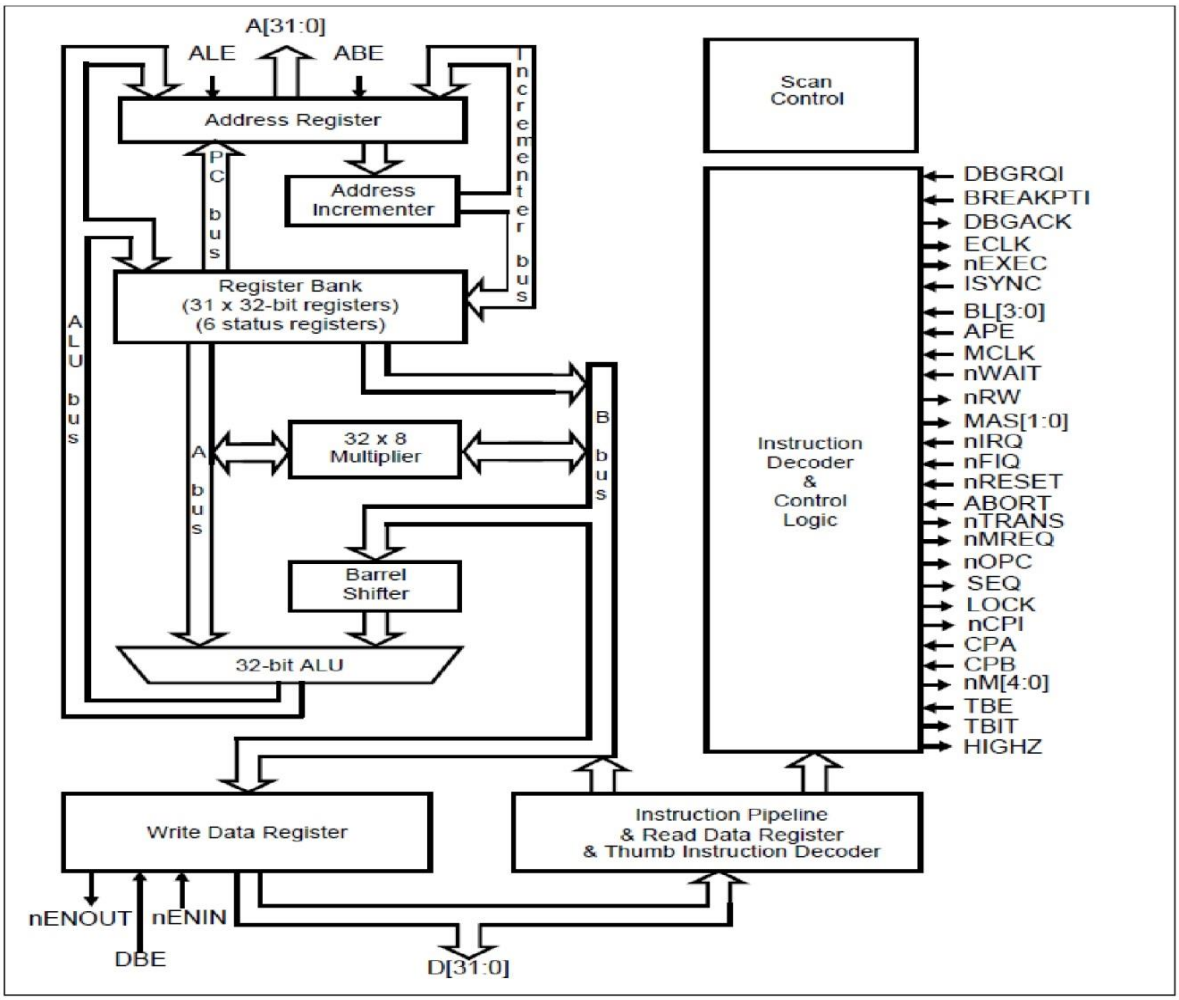


Fig3.1 ARM7 TDMI Core Diagram

3.1.2 Instruction Pipeline

The ARM7TDMI core uses a three-stage pipeline to increase the flow of instructions to the processor. This allows multiple simultaneous operations to take place and continuous operation of the processing and memory systems.

The instructions are executed in three stages:

- fetch
- decode
- execute

3.1.2.1 Memory Interface

The ARM7TDMI memory interface is designed to allow optimum performance potential and minimize memory usage. Speed critical control signals are pipelined to allow system control functions to exploit the fast-burst access modes supported by many memory technologies. The ARM7TDMI has four basic types of memory cycle:

- Internal
- Non sequential
- Sequential
- Coprocessor registers transfer.

There is also the option to use either a single bidirectional data bus or two separate unidirectional data input and output buses.

3.1.2.2 Memory Formats

The ARM7TDMI can be configured to treat stored words in either big-endian or little-endian format.

3.1.2.3 Performance, Code Density And Operating States

The ARM7TDMI core supports two operating states and instruction sets:

- ARM state for 32-bit, word-aligned instructions
- Thumb state for 16-bit, half word-aligned instructions.

The ARM instruction set allows a program to achieve maximum performance with the minimum number of instructions. The simpler thumb instruction set offers much increased code density reducing memory requirement. Code can switch between the ARM and thumb instruction sets on any procedure call.

3.1.3 Operating Modes

The ARM7TDMI core has seven modes of operation:

- User mode is the usual program execution state
- Fast Interrupt (FIQ) mode supports data transfer or channel processes to allow very fast interrupt processing and to preserve values across interrupt calls
- Interrupt (IRQ) mode is used for general purpose interrupt handling
- Supervisor mode is a protected mode for the operating system
- Abort mode is entered after a data or instruction pre fetch abort
- System mode is a privileged user mode for the operating system
- Undefined mode is entered when an undefined instruction is executed.

3.1.3.1 Coprocessors

Up to 16 coprocessors can be connected to an ARM7TDMI system.

3.1.3.2 Debug Features

Internal state of the ARM core can be examined using a JTAG interface to allow the insertion of instructions into core pipeline and avoid using external data bus. ARM7TDMI core includes an internal functional unit known as the Embedded ICE logic. The Embedded ICE logic is configured to monitor the ARM7TDMI core actively for specific instruction fetches and data accesses.

3.1.4 ARM7TDMI Processor Core

The ARM7TDMI processor core implements the ARMv4T Instruction Set Architecture (ISA).This is a superset of the ARMv4 ISA which adds support for the 16-bit Thumb instruction set. Software using the Thumb instruction set is compatible with all members of the ARM Thumb family, including ARM9, ARM9E, and ARM10 families.

3.1.4.1 Registers

The ARM7TDMI core consists of a 32-bit data path and associated control logic. This data path contains 31 general-purpose 32-bit registers, 7 dedicated 32-bit registers coupled to a barrel-shifter, Arithmetic Logic Unit, and multiplier.

3.1.4.2 Modes and exceptions

The ARM7TDMI supports seven modes of operation:

- User mode
- Fast Interrupt (FIQ)
- Interrupt (IRQ)
- Supervisor mode
- Abort mode

• Undefined mode and System mode.

All modes other than User are privileged modes. These are used to service hardware interrupts, exceptions, and software interrupts. Each privileged mode has an associated Saved Program Status Register (SPSR). This register is use to save the state of the Current Program Status Register (CPSR) of the task immediately before the exception occurs. In these privileged modes, mode-specific banked registers are available. These are automatically restored to their original values on return to the previous mode and the saved CPSR restored from the SPSR.

System mode does not have any banked registers. It uses the User mode registers. System mode runs tasks that require a privileged processor mode and allows them to invoke all classes of exception.

3.1.4.3 Processor States

The ARM7TDMI processor can be in one of two states:

- ARM state
- In ARM state, 16 general registers and one or two status registers are accessible at any one time. The ARM state register set contains 16 directly accessible registers: R0 to R15. All of these except R15 are general-purpose, and may be used to hold either data or address values.

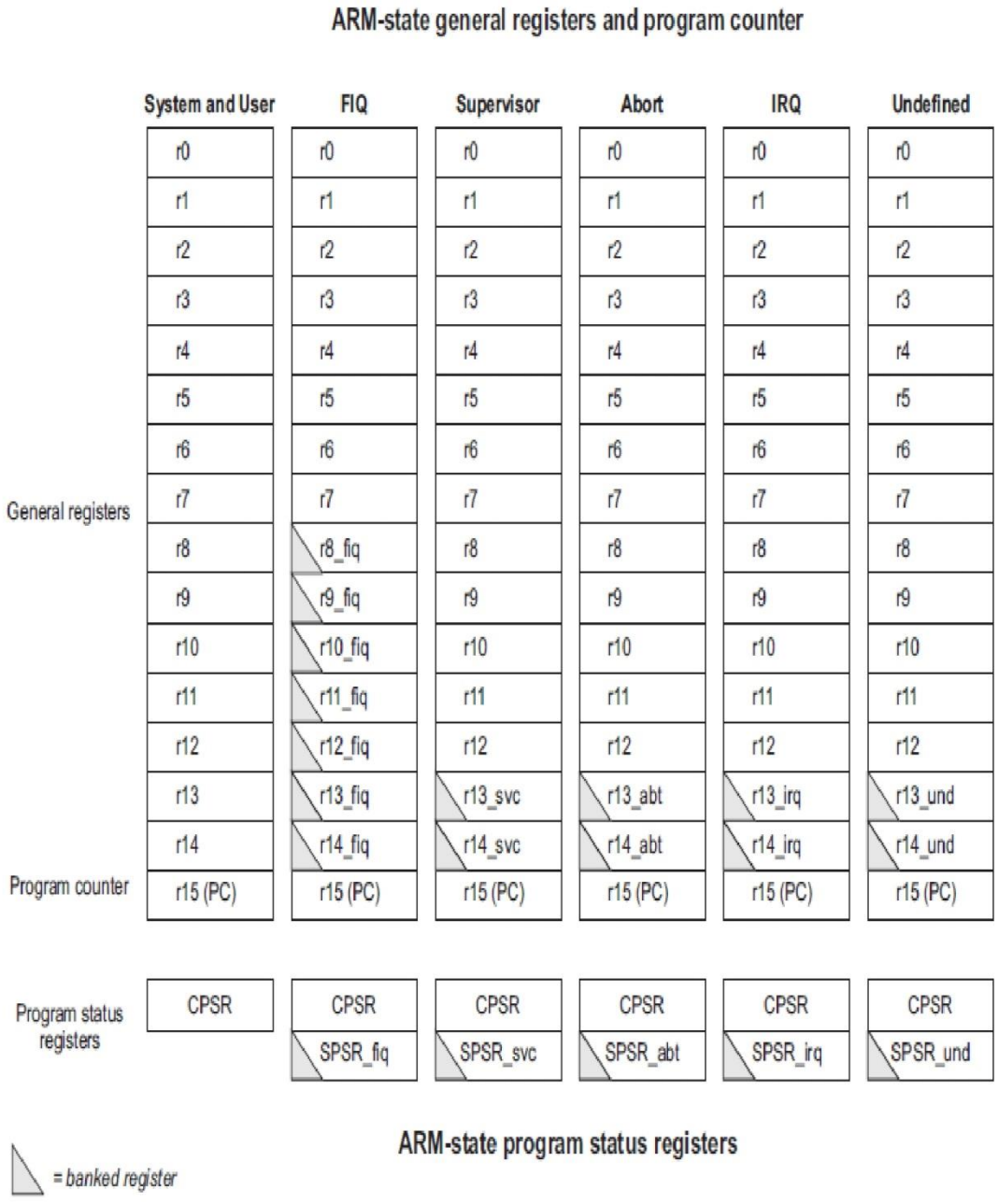


Fig3.2 Register Organization in ARM state

.

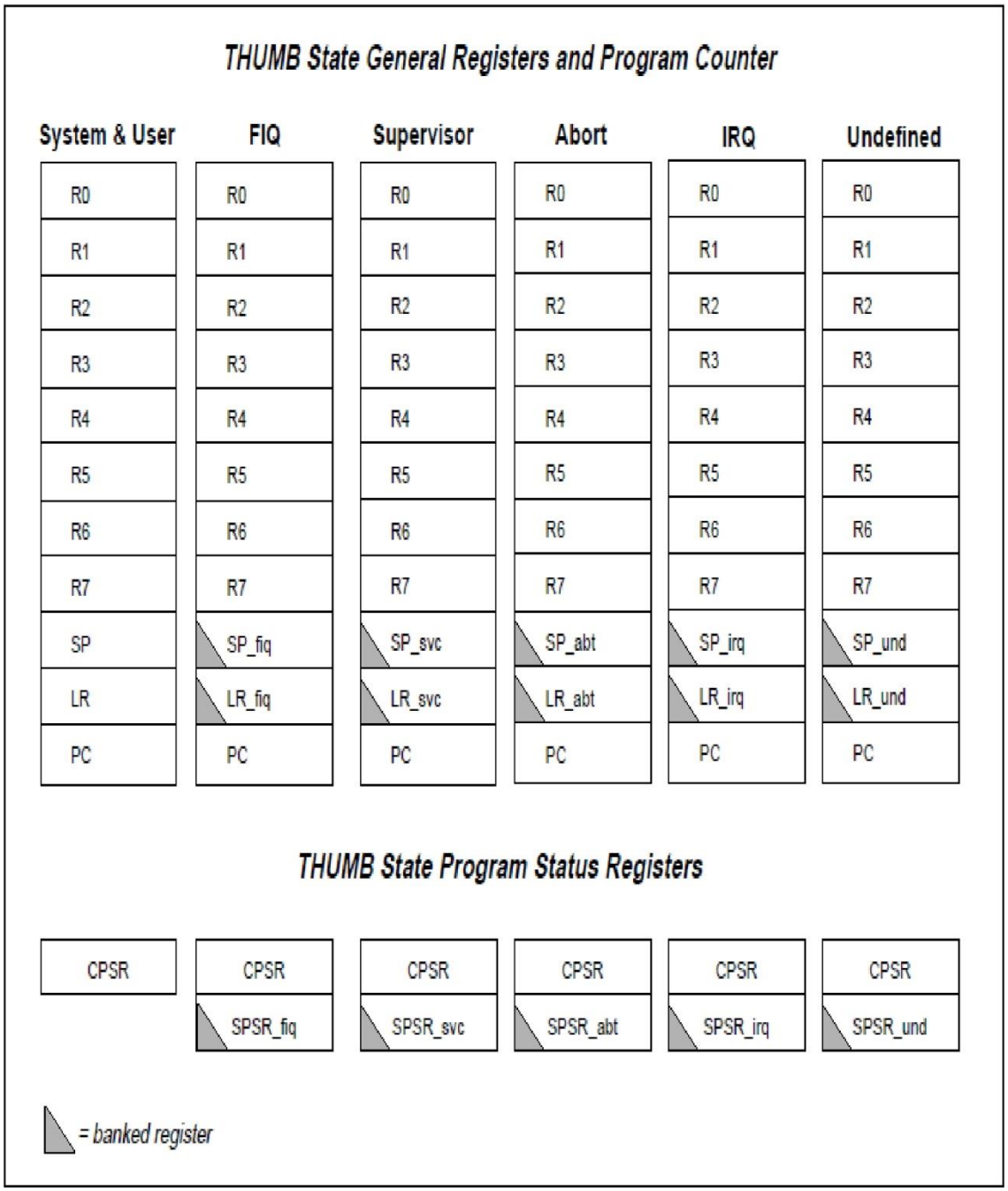


Fig3.3 Register Organization in THUMB state

The THUMB state register set is a subset of the ARM state set. The programmer has direct access to eight general registers, R0-R7, as well as the Program Counter (PC), a stack pointer register (SP), a link register (LR), and the CPSR. There are banked Stack Pointers, Link Registers and Saved Process Status Registers (SPSRs) for each privileged mode.

The registers available to the programmer in each mode, in THUMB state, are illustrated in Figure 2.3.3.2 Register Organization in THUMB state.

3.1.4.4 Exceptions

The ARM7TDMI supports seven types of exception:

- FIQ – fast interrupt
- IRQ – normal interrupt
- Data abort
- Pre fetch abort
- Software interrupt
- Undefined instruction
- Reset.

All exceptions have banked registers for R14 and R13. After an exception, R14 holds the return address for exception processing. This address is used both to return after the exception is processed and to address the instruction that caused the exception. R13 is banked across exception modes to provide each exception handler with a private stack pointer. The fast interrupt mode also banks registers 8 to 12 so that interrupt processing can begin without the need to save or restore these registers.

3.1.4.5 Status Registers

All other processor states are held in status registers. The current operating processor status in the CPSR. The CPSR holds:

- Four ALU flags (Negative, Zero, Carry, and Overflow)
- An interrupt disable bit for each of the FIQ and IRQ interrupts
- A bit to indicate ARM or Thumb execution state

Five bits to encode the current processor mode.

3.1.5 Lpc2148 Microcontroller

LPC2148 microcontroller board based on a 16-bit/32-bit ARM7TDMI-S CPU with real-time emulation and embedded trace support, that combine microcontrollers with embedded high-speed flash memory ranging from 32 kB to 512 kB. A 128-bit wide memory interface and unique accelerator architecture enable 32-bit code execution at the maximum clock rate. For critical code size applications, the alternative 16-bit Thumb mode reduces code by more than 30% with minimal performance penalty. The meaning of LPC is Low Power Low Cost microcontroller. This is 32 bit microcontroller manufactured by Philips semiconductors (NXP).

Due to their tiny size and low power consumption, LPC2148 is ideal for applications where miniaturization is a key requirement, such as access control and point-of-sale.

3.1.5.1 LPC2148 Microcontroller Architecture

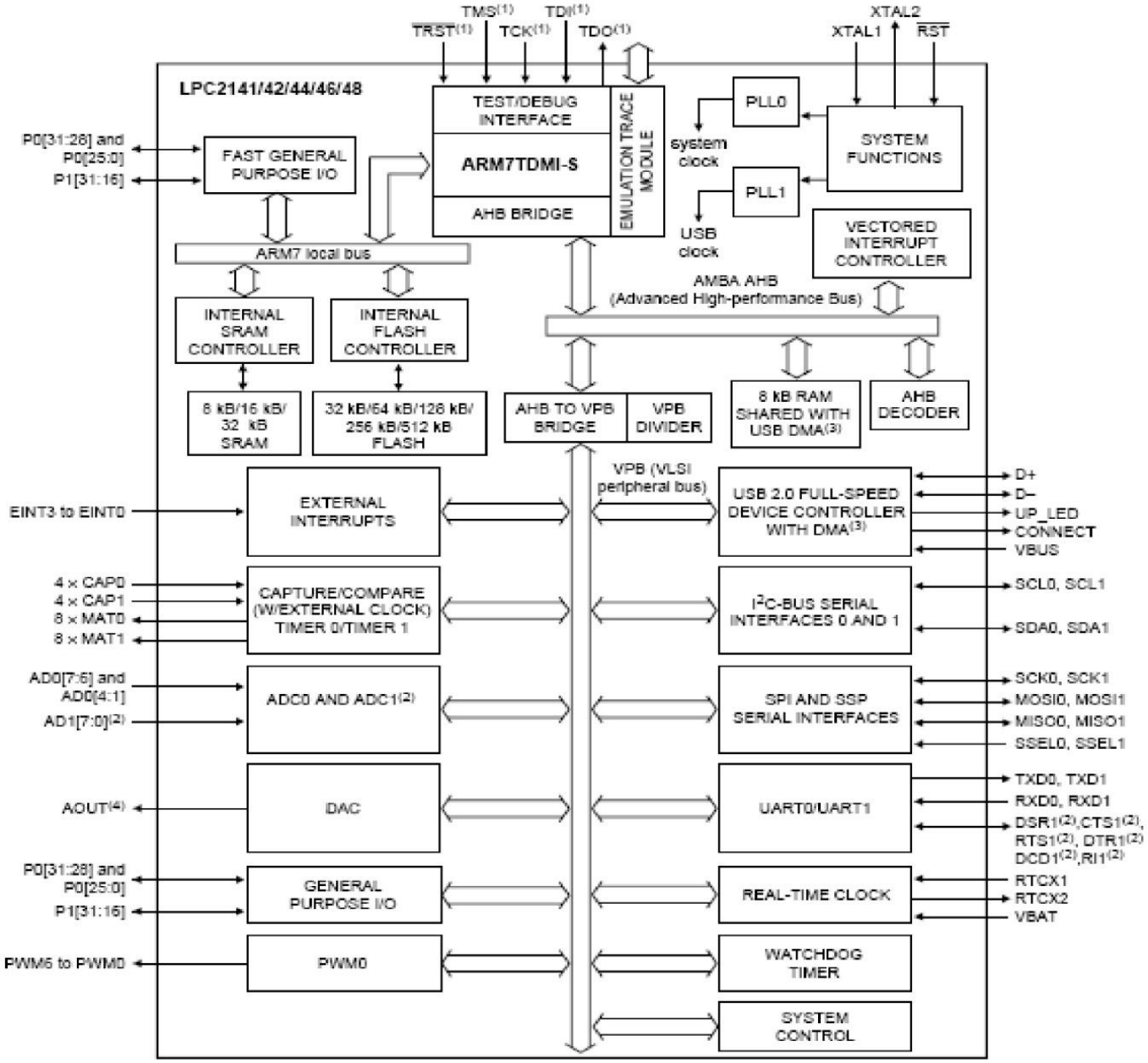


Fig 3.4 LPC2148 Microcontroller Architecture

3.1.5.2 Pin Diagram

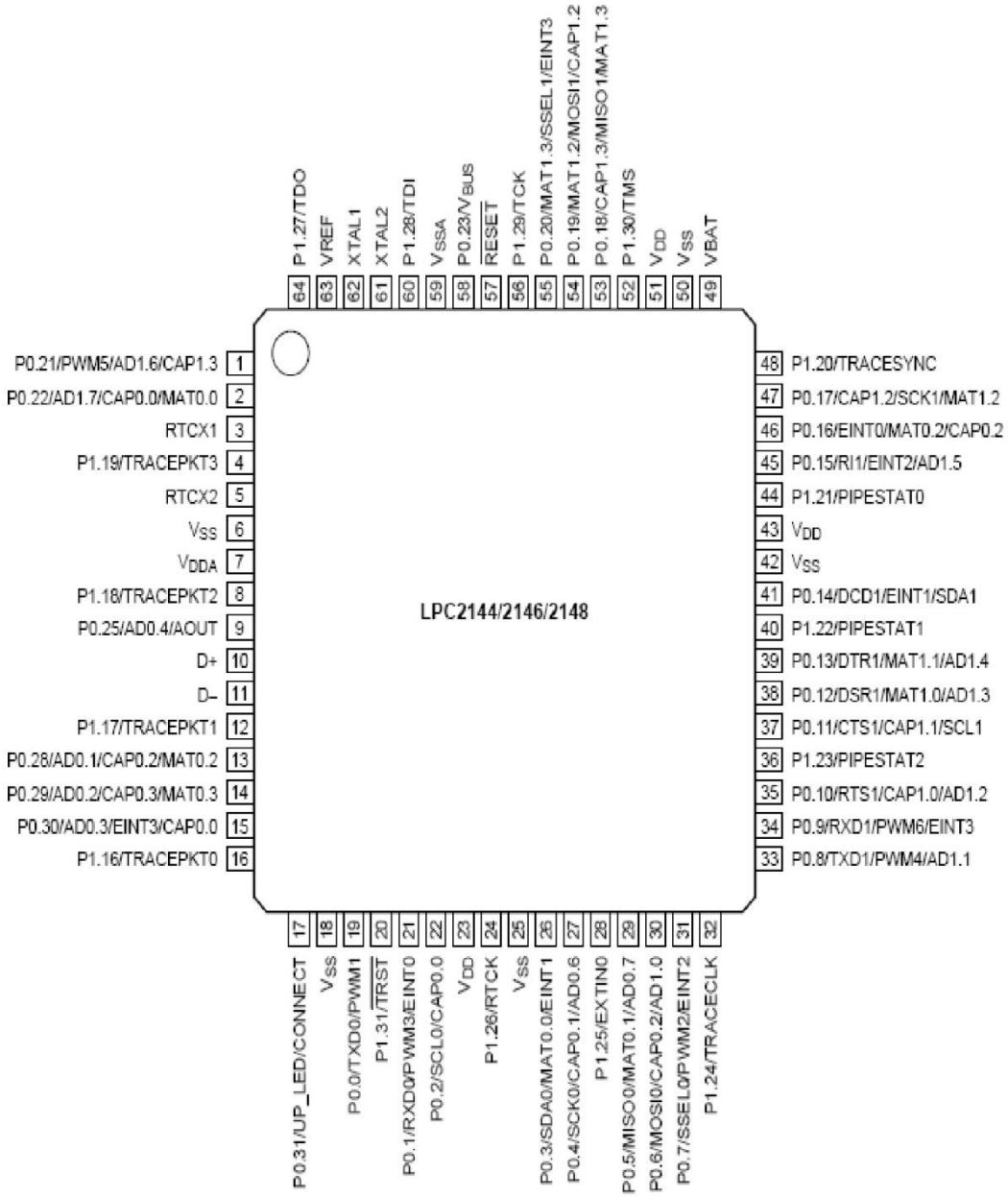


Fig 3.5 LPC2148 Microcontroller Pin Diagram

3.1.5.3 Architectural Overview

The ARM7TDMI-S is a general purpose 32-bit microprocessor, which offers high performance and very low power consumption. The ARM architecture is based on Reduced Instruction Set Computer (RISC) principles, and the instruction set and related decode mechanism are much simpler than those of micro programmed Complex Instruction Set Computers (CISC). This simplicity results in a high instruction throughput and impressive real-time interrupt response from a small and cost-effective processor core.

Pipeline techniques are employed so that all parts of the processing and memory systems can operate continuously. Typically, while one instruction is being executed, its successor is being decoded, and a third instruction is being fetched from memory. The ARM7TDMI-S processor also employs a unique architectural strategy known as Thumb, which makes it ideally suited to high-volume applications with memory restrictions, or applications where code density is an issue.

The key idea behind Thumb is that of a super-reduced instruction set. Essentially, the ARM7TDMI-S processor has two instruction sets:

- The standard 32-bit ARM set.
- A 16-bit Thumb set.

The Thumb set's 16-bit instruction length allows it to approach twice the density of standard ARM code while retaining most of the ARM's performance advantage over a traditional 16-bit processor using 16-bit registers. This is possible because Thumb code operates on the same 32-bit register set as ARM code. Thumb code is able to provide up to 65 % of the code size of ARM, and 160 % of the performance of an equivalent ARM processor connected to a 16-bit memory

system. The particular flash implementation in the LPC2141/42/44/46/48 allows for full speed execution also in ARM mode. It is recommended to program performance critical and short code sections (such as interrupt service routines and DSP algorithms) in ARM mode. The impact on the overall code size will be minimal but the speed can be increased by 30% over Thumb mode.

3.2 Controller Area Network (CAN)

3.2.1 Need For CAN

CAN protocol is a method of communication between various electronic devices like engine management systems, active suspension, ABS, gear control, lighting control, air conditioning, airbags, central locking etc embedded in an automobile. An idea initiated by Robert Bosch GmbH in 1983 to improve the quality of automobiles thereby making them more reliable, safe and fuel efficient. The increasing number of electronic devices used communication signals with more complex interrelations between them. Thereby making the life difficult for automobile engineers when they designed systems wherein one electronic device needs to communicate with others to operate.

A vehicle contains a network of electronic devices to share information/data with each other. It uses information from engine control unit and various sensors in the system. Before CAN was introduced in the automobiles, each electronic device was connected to other device using the wires (point to

point wiring) which worked fine, when the functions in the system were limited.

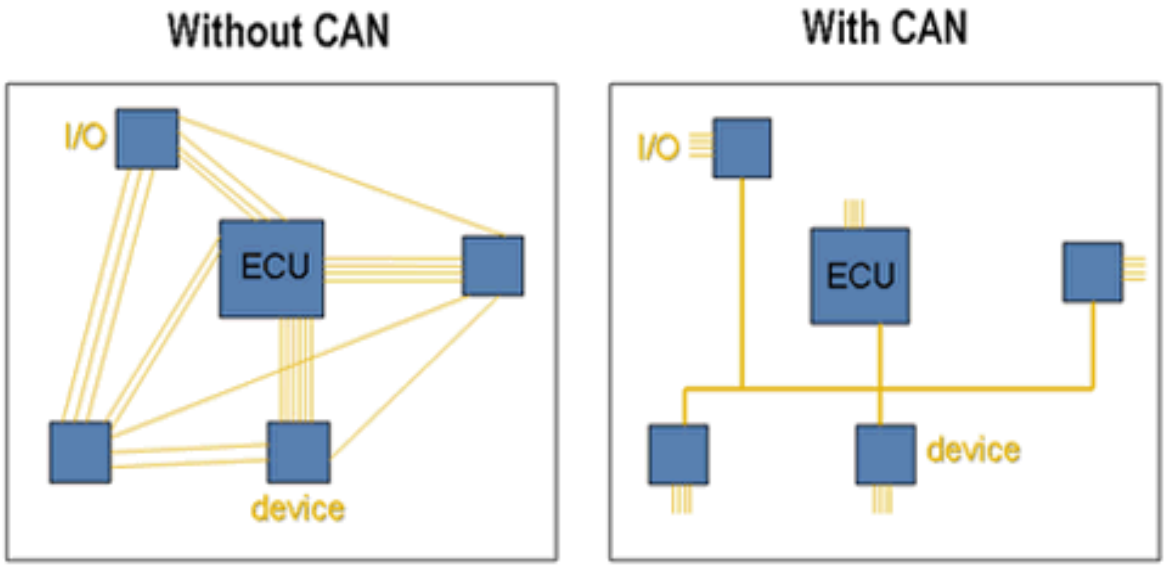


Fig3.6 Structure of CAN

3.2.2 Can Message Frame

A CAN communication network can be configured different frame or message formats. Standard or Base Frame Format or CAN 2.0 A and Extended Frame Format or CAN 2.0 B.

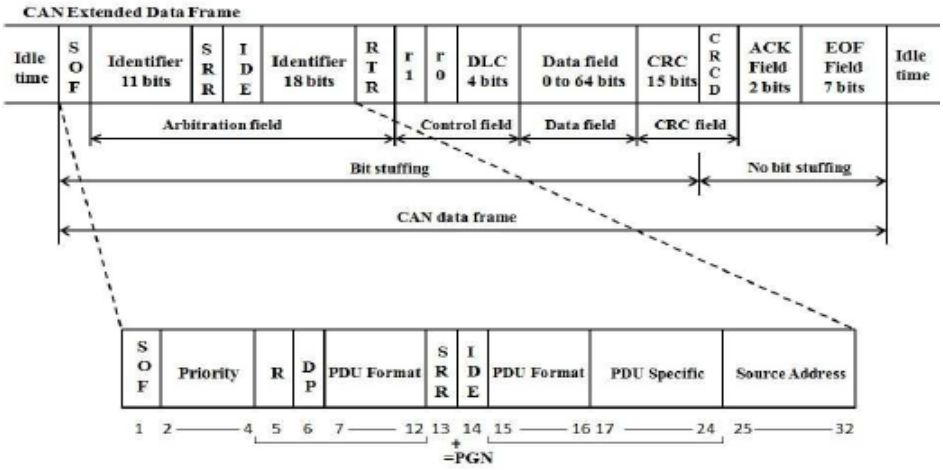


Fig3.7 Message Frame

The difference between these two formats is that the length of bits, i.e., the base frame supports 11-bits length for the identifier, whereas the extended frame supports 29-bits length for the identifier, which is made up of 18-bit extension and an 11-bit identifier. The IDE bit differs CAN extended frame format and the CAN base frame format wherein IDE is transmitted as dominant in an11-bit frame case and recessive in a 29-bit frame case. It is also possible to send or receive messages in base frame format by some CAN controllers that support extended-frame formats.

3.2.3 Priority

Every message has a priority, so if two nodes try to send messages simultaneously, the one with the higher priority gets transmitted and the one with the lower priority gets postponed. This arbitration is non-destructive and results in non-interrupted transmission of the highest priority message. This also allows networks to meet deterministic timing constraints.

3.3 Sensors

3.3.1 Ultrasonic sensor

Ultrasonic sensor is a device that can measure the distance to an object by using sound waves. It measures distance by sending out a sound wave at a specific frequency and listening for that sound wave to bounce back. By recording the elapsed time between the sound wave being generated and the sound wave bouncing back, it is possible to calculate the distance between the sonar sensor and the object.



Fig3.8 Ultrasonic sensor

HC-SR04 Ultrasonic Distance Sensor is a popular and low cost solution for non-contact distance measurement function. It is able to measure distances from 2cm to 400cm with an accuracy of about 3mm. This module includes ultrasonic transmitter, ultrasonic receiver and its control circuit.

3.3.2 Temperature Sensor

The LM35 series are precision integrated-circuit temperature sensors, whose output voltage is linearly proportional to the Celsius (Centigrade) temperature. The LM35 thus has an advantage over linear temperature sensors calibrated in ° Kelvin, as the user is not required to subtract a large constant voltage from its output to obtain convenient Centigrade scaling. The LM35 does not require any external calibration or trimming to provide typical accuracies of ±1⁄4°C at room temperature and ±3⁄4°C over a full −55 to +150°C temperature range.

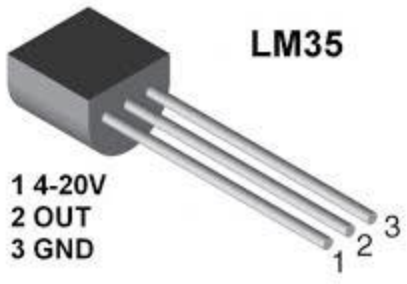


Fig3.9 Temperature sensor

Low cost is assured by trimming and calibration at the wafer level. The LM35's low output impedance, linear output, and precise inherent calibration make interfacing to readout or control circuitry especially easy. It can be used with single power supplies, or with plus and minus supplies. As it draws only 60 μA from its supply, it has very low self-heating, less than 0.1°C in still air. The LM35 is rated to operate over a −55° to +150°C temperature range, while the LM35C is rated for a −40° to +110°C range (−10° with improved accuracy). The LM35 series is available packaged in hermetic TO-46 transistor packages, while the LM35C, LM35CA, and LM35D are also available in the plastic TO-92 transistor package. The LM35D is also available in an 8-lead surface mount small outline package and a plastic TO-220 package.

3.3.2.1 Features

- Calibrated directly in ° Celsius (Centigrade)
- Linear + 10.0 mV/°C scale factor
- 0.5°C accuracy guarantee able (at +25°C)
- Rated for full −55° to +150°C range

- Suitable for remote applications
- Low cost due to wafer-level trimming
- Operates from 4 to 30 volts
- Less than 60 μA current drain
- Low self-heating, 0.08°C in still air
- Nonlinearity only ±1⁄4°C typical
- Low impedance output, 0.1 □ □ for 1 mA load

3.3.3 Fuel Level Sensor

Level sensors detect the level of liquids and other fluids and fluidized solids, including slurries, granular materials, and powders that exhibit an upper free surface.



Fig 3.10 Fuel level sensor

Water sensor brick is designed for water detection, which can be widely used in sensing rainfall, water level, and even liquid leakage. Connecting a water sensor to an ARM 7 controller is a great way to detect a leak, spill, flood, rain, etc. It can be used to detect the presence, the level, the volume and/or the absence of water. While this could be used to remind you to water your plants, there is a better Grove

sensor for that. The sensor has an array of exposed traces, which read LOW when water is detected.

3.3.4 Pressure Sensor

A pressure sensor is a device equipped with a pressure-sensitive element that measures the pressure of a tire or a liquid against a diaphragm made of stainless steel, silicon, etc., and converts the measured value into an electrical signal as an output.

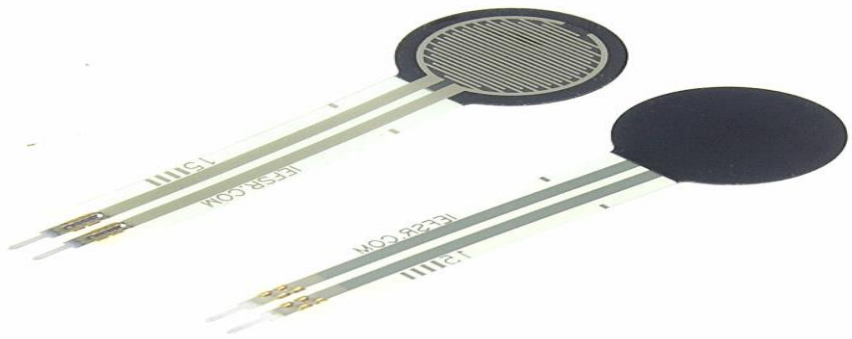


Fig 3.11.Pressure sensor

There are sensors that measure the absolute pressure and those that measure the pressure relative to atmospheric pressure or a specified pressure. For sensors that use atmospheric pressure as a reference, there are sensors that measure negative pressures and positive pressures. A semiconductor piezo-resistance dispersion pressure sensor has a semiconductor distortion gauge formed on the surface of the diaphragm, and it converts changes in electrical resistance into an

electrical signal by means of the piezo-resistance effect that occurs when the diaphragm is distorted due to an external force (pressure).

3.4 LCD Module (2x16 Character)

Dot matrix LCD modules is used for display the parameters and fault condition.16 characters 2 lines display is used. It has controller which interface data's and LCD panel. Liquid crystal displays (LCD's) have materials, which combine the properties of both liquids and crystals. Rather than having a melting point, they have a temperature range within which the molecules are almost as mobile as they would be in a liquid, but are grouped together in an ordered form similar to a crystal. An LCD consists of two glass panels, with the liquid crystal material sandwiched in between them. The inner surface of the glass plates are coated with transparent electrodes which define the character, symbols or patterns to be displayed polymeric layers are present in between the electrodes and the liquid crystal molecules to maintain a defined orientation angle.

One each polarizer's are pasted outside the two glass panels. These polarizer's would rotate the light rays passing through them to a definite angle, in a particular direction When the LCD is in the off state, light rays are rotated by the two polarizes and the liquid crystal, such that the light rays come out of the LCD without any orientation, and hence the LCD appears transparent.

When sufficient voltage is applied to the electrodes, the liquid crystal molecules would be aligned on a specific direction. The light rays passing through the LCD would be rotated by the polarizes, which would result in activating/highlighting the desired characters.



Fig3.12 LCD Display

The LCD's are lightweight with only a few millimeters thickness. since the LCD's consume less power, they are compatible with low power electronic circuits, and can be powered for long durations .The LCD's don't generate light is needed to read the display. By using backlighting, reading is possible in the dark .The LCD's have long life and a wide operating temperature range.

One of the most popular output devices for embedded electronics is LCD. The LCD interface has become very simple. This is due to the availability modules for LCDs. The LCD along with necessary controller (LCD Controller) and mounting facility is made available in the module itself. The LCD controller takes care of everything necessary for the LCD. We communicate with the LCD controller with the help of a command set provided by the manufacturer.

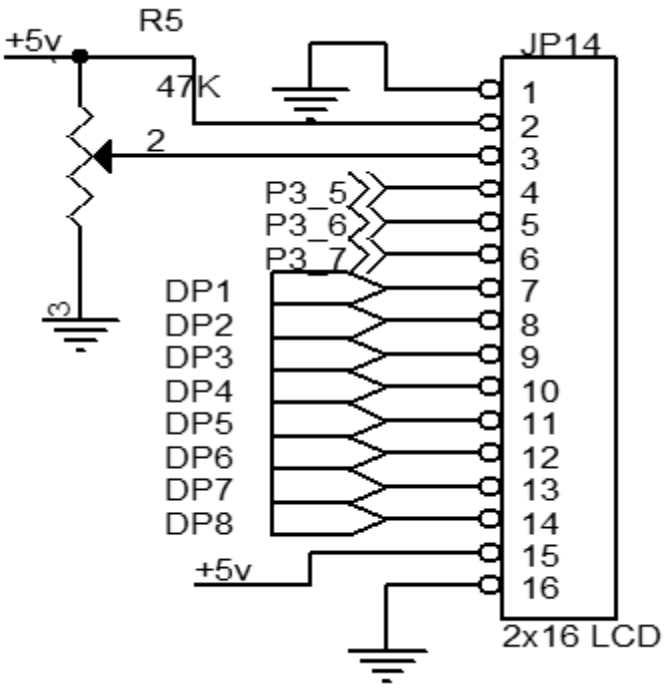


Fig3.13 circuit diagram

This circuit consists of a Microcontroller and a LCD. This LCD is operating with an 8-bit data bus. So totally 11 data lines are required (8 Data lines and 3 control lines). The 8 bit data lines are connected to the Port1 and the 3 control lines to the Port3.5-Port3.7. The EN line is called "Enable." This control line indicates to the LCD that we are sending it data. To send data to the LCD, the EN should be low (0) and then set the other two control lines and/or put data on the data bus. When the other lines are completely ready, bring EN high (1) and wait for the minimum amount of time required by the LCD datasheet (this varies from LCD to LCD), and end by bringing it low (0) again.

The RS line is the "Register Select" line. When RS is low (0), the data is to be treated as a command or special instruction (such as clear screen, position

cursor, etc.). When RS is high (1), the data being sent is text data, which should be displayed on the screen. For example, to display the letter "T" on the screen we would set RS high.

The RW line is the "Read/Write" control line. When RW is low (0), the information on the data bus is being written to the LCD. When RW is high (1), the program is effectively querying (or reading) the LCD. Only one instruction ("Get LCD status") is a read command. All others are written commands so RW will almost always be low.

CHAPTER 4

SOFTWARE TOOLS

4.1 Keil C

Keil software is the leading vendor for 8/16-bit development tools (ranked at first position in the 2004 embedded market study of the embedded system and EE times magazine). Keil software is represented world wide in more than 40 countries, since the market introduction in 1988; the keil C51 compiler is the de facto industry standard and supports more than 500 current 8051 device variants. Now, keil software offers development tools for ARM. Keil software makes C compilers, macro assemblers, real-time kernels, debuggers, simulators, integrated environments, and evaluation boards for 8051, 251, ARM and XC16x/C16x/ST10 microcontroller families.

The Keil C51 C Compiler for the 8051 microcontroller is the most popular 8051 C compiler in the world. It provides more features than any other 8051 C compiler available today. The C51 Compiler allows you to write 8051 microcontroller applications in C that, once compiled, have the efficiency and speed of assembly language. Language extensions in the C51 Compiler give you full access to all resources of the 8051. The C51 Compiler translates C source files into relocatable object modules which contain full symbolic information for debugging with the µVision Debugger or an in-circuit emulator. In addition to the object file, the compiler generates a listing file which may optionally include symbol table and cross reference

- Nine basic data types, including 32-bit IEEE floating-point,

- Flexible variable allocation with bit, data, bdata, idata, xdata, and pdata memory types,
- Interrupt functions may be written in C,
- Full use of the 8051 register banks,
- Complete symbol and type information for source-level debugging,
- Use of AJMP and ACALL instructions,
- Bit-addressable data objects,
- Built-in interface for the RTX51 real time kernels,
- Support for the Philips 8*x*C750, 8*x*C751, and 8*x*C752 limited instruction sets,
- Support for the Infineon 80C517 arithmetic unit.

4.2 FLASH MAGIC

Flash magic can control the entry into ISP mode of some microcontroller devices by using the COM port handshaking signals to control the device. Typically the handshaking signals are used to control such pins as Reset, PSEN and VCC. The exact pins used depend on the specific device.

When this feature is supported, Flash Magic will automatically place the device into ISP mode at the beginning of an ISP operation. Flash Magic will then automatically cause the device to execute code at the end of the ISP operation.

CHAPTER 5

PHOTOGRAPH OF WORKING MODEL

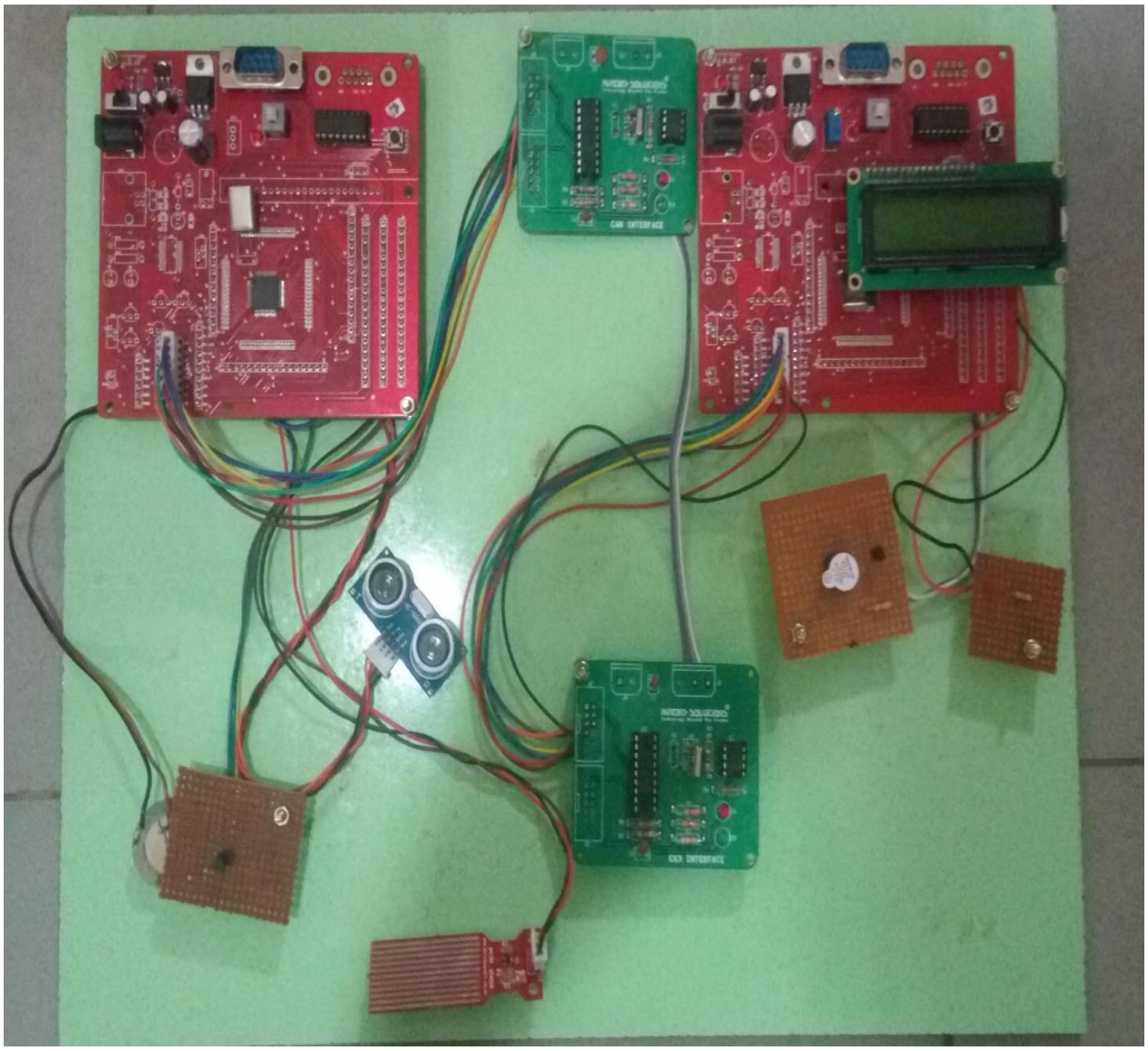


Fig 5.1 photograph of working model

CHAPTER 6

WORKING PRINCPLE

Our project automotive safety systems is based on establishing communication between sensors, objects and engine for the vehicle is interfaced with the arm7. Our project is a CAN based architecture designed for the purpose of Vehicular health monitoring. CAN is a Message based protocol designed specifically for Automotive, later Aerospace, Industrial automation and Medical equipment's. CAN interface module is used to communicate the monitored parameters between the vehicle parts and the control center. During the transmission of the data from one node to another node disturbance occurs. To avoid these disturbances we propose CAN protocol. In this project we declare the system with ARM and CAN protocol to monitor and diagnose the problems in the transportation application.

The project deals with the data transmission between two units in the exact time without any disturbance. The data transmission time is increased with the CAN protocol. Here vehicles health condition like temperature, fuel level and air pressure are measured by using temperature, pressure and level sensor. The vehicle collision can be avoided by using the ultrasonic sensor. All the sensors are connected with the acquisition node and these details are send to the processing node through CAN transceiver. Whenever the sensors value reaches abnormal stages, the alert will be generated with the help of alarm and these details are displayed in the LCD in the drivers section.

CHAPTER 7

CONCLUSION

This project introduces an embedded system with a combination of CAN bus systems. Digital control of the vehicle is an important criterion of modern technology. With the rapid development of embedded technology, high performance embedded processor is penetrated into the automotive industry, which is low cost, high reliability and other features to meet the needs of the modern automobile industry. The proposed high-speed CAN bus system solves the problem of automotive system applications, also has a certain practical value and significance. With PIC as the main controller and it makes full use of the high-performance of PIC18f458, high-speed reduction of CAN bus communication control networks and instrument control. so ,as to achieve full sharing of data between nodes and enhance their collaborative work. This system features efficient data transfer among different nodes in the practical applications. And it introduced the distance measurement using ultrasonic sensors. This proposed system provides the intelligent breaking system using RFID module at automatic braking system enabled vehicles.

REFERENCE

[1] Artal J.S., Caraballo J. and Dufo R. *"CAN/LIN-Bus Protocol. Implementation of a low-cost serial communication network"* 978-1-4799-6002-6/14/$31.00 ©2014 IEEE.

[2] B.Gmbh, "*CAN specification*", vol 1 Version 2.0, 1991.

[3] Pazul, "*Controller Area Network (CAN) Basics*", Microchip technology, AN713, May 1999.

[4] Wilfried Voss, '*A comprehensive guide to controller area network*", Copperhill Media Corporation, 2005-2008.

[5] Benjamin C Kuo, M. Farid Golnaraghi, "*Automatic Control systems, Eight edition*", John wiley & sons., Inc 2003.