ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 9, September 2013

Industrial Manifold Control Systems Using High Speed Networks

S. RAJESWARI[1], S.JASMINE[2]

Assistant professor, ECE Dept, Bharath University, Chennai[1, 2]

ABSTRACT: In this paper, a multi-motor control circuit based on controller area network (CAN) bus is presented. The PIC (16f877A) microcontroller is the foundation and the circuit which is based on the usage of inbuilt digital-to-Analog converter, in order to achieve the breakdown of motor control. In the subdivision control, subdivision steps are set by software. Article details the overall structure of the system of hardware design, software design process, and CAN bus controller structure. Compared with the traditional centralized control, the multi-motor control system has higher reliability. Experimental result shows that the system and the intelligent control module is effective and efficient..

KEYWORDS: PIC Microcontroller 16F877A, CAN controller.

I. INTRODUCTION

Motor is use in many industries for various purpose which acts as a bridge between man and machine. Even though we use different types of motor we need a control unit to control motor. Controlling plays a vital role in industries. Typically industries may use hundreds of motor and hence they need a individual control unit for each motor.

In exiting systems each motor is provided with individual motor control unit or same speed operating motors needs a single control unit. Due to the complexity of controlling unit the cost, complexity, maintenance and increase in wiring harness which cause difficulty in error finding and debugging. For this type of controlling network the control unit must be placed near the motor itself. To get an efficient operation of motor cannot be achieved by controlling unit because in a large number of motor networks the fault cannot be identify easily which may be a physical fault or anything else. For achieving a controlled, proper and effective operation of motor speed control controller, with embedded system technology have used.

PIC controller is a short form of “Peripheral Interface Controller” which is used to interfacing the input/output devices with control unit. Controller area Network (CAN) is an ISO standard protocol to get reliable and fast information transmission. By using this protocol the input signal can be sent without any losses to motor even though long distances (upto 6Km). After that the main advantage of CAN protocol that having proven reliability, high speed, error handling and error confinement.

The controller area network (CAN) was originally developed by Bosch in the early'80s. In order to solve the vehicle control and measure a large number of data exchange between devices and the development of a serial data communication bus, belonging to the field bus area and it became to be international standards (ISOH898: road vehicles, high-speed digital switching system Controller Area Kong Standard) in 1993. CAN bus is an abbreviation; the full name should be "Controller Area Network Bus" Controller Area Network is in English the first letter combination. It is a bus, with our common USB bus belongs to a class concept, but CAN bus using differential signal transmission, have a strong ability of error detection, communication distance, was used in some special occasions, such as cars, factories and mines, such as strong local interference.

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 9, September 2013

II. BASIC DESPCRIPTION OF THE MODULE

The entire operation of our project is shown in the form of block diagram thereby while seeing that we can easily understand the aim and operation of the project. The block diagram consist of Power supply, Microcontroller, CAN Controller, CAN Transceiver, CAN bus and motor.

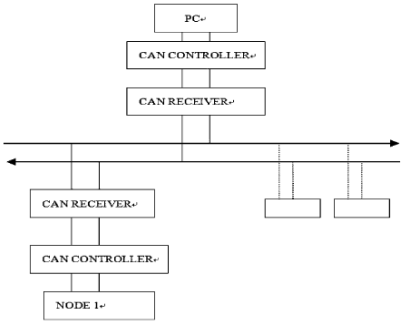


Fig 1 : CAN BUS BASIC DESCIPRTION

The power supply is given to every unit such as PIC controller, CAN controller, CAN transceiver and Motor driver for its operation. CAN bus is a communication medium between control unit and motor network. The PIC controller plays a vital role in this system. It is interfaced with CAN controller, LCD display and motor driver. The data is transmitted to motor network by serial communication via CAN bus.

CAN controller and CAN transceiver takes main role for data transmission with efficiency. The output from the CAN controller is given to CAN transceiver where the signal is transmitted as differential output. CAN bus is a two wire differential bus which data transmission capability about 1Mbits per second. When the power supply is turned on by varying the potentiometers the speed of motor can be controlled with the help of PIC, CAN controller, CAN transceiver and CAN bus.

Every circuit should have a power supply for working. So that the power supply is the main requirement for the working of devices. All the electronic components starting from diode to Intel IC's only work with a DC supply ranging from ±5v to ±12v. We are utilizing for the same, the most cheaply and commonly energy source of 230V-50Hz and stepping down, rectifying, filtering and regulating the voltage.

ECU (Embedded Control Unit) is the heart of this project. It will control overall operation of this system. The signal from the motor is fed to this control unit and it will process the data or signal and it will start the operation and also the processed data will be sent to the LCD for monitoring purpose. This unit uses PIC16F877A Microcontroller for this operation.

Controller Area Network (CAN) protocol controller implementing CAN specification V2.0 A/B. It supports CAN 1.2, CAN 2.0A, CAN 2.0B Passive, and CAN 2.0B Active versions of the protocol, and is capable of transmitting and receiving standard and extended messages. It is also capable of both acceptance filtering and message management. It includes three transmit buffers and two receive buffers that reduce the amount of microcontroller (MCU) management required. The MCU communication is implemented via an industry standard Serial Peripheral Interface (SPI) with data rates up to 5 Mb/s.

The CAN bus is a balanced (differential) 2-wire interface running wires. The Bit Encoding used is Non Return to Zero (NRZ) encoding (with bit-stuffing) for data communication on a differential two wire bus. The use of NRZ encoding ensures compact messages with a minimum number of transitions and high resilience to external disturbance. The maximum line length is 1Km, 40 meters at 1Mbps.

The L293D is designed to provide bidirectional drive currents up to 600-mA at voltages from 4.5V to 36V. This device have designed mainly to drive inductive loads such as relays, solenoids, dc and bipolar stepping motors, as well as other high-current/high-voltage loads in positive-supply applications.

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 9, September 2013

A. Hardware Used

PIC MICROCONTROLLER 16F877A:

PIC16F877A microcontroller which is popular at this moment starts from beginner until all professionals. Because of the operation of PIC16F877A is easy and by using its FLASH memory technology write-erase can be possible until one lakh of times. PIC16F877A have 40 pin by 32 path of I/O.

PIC16F877A perfectly fits many uses, from automotive industries and controlling home appliances to industrial instruments, remote sensors, electrical door locks and safety devices. It is also ideal for smart cards as well as for battery supplied devices because of its low power consumption. EEPROM memory makes it easier to apply microcontrollers to devices where permanent storage of various parameters is needed (codes for transmitters, motor speed, receiver frequencies, etc.). In System Programmability of this chip (along with using only two pins in data transfer) makes possible the flexibility of a product, after assembling and testing have been completed. This capability can be used to create assembly-line production, to store calibration data available only after final testing, or it can be used to improve programs on finished products

Normally all input signals are in analog form in electrical circuit but microcontroller can understand the digital inputs only i.e. zeros and ones. For the conversion of analog to digital, we are using ADC but in PIC 16F877A ADC is an inbuilt feature. ADC is responsible for converting information about some analog value to a binary number and for follows it through to a CPU block for further process.

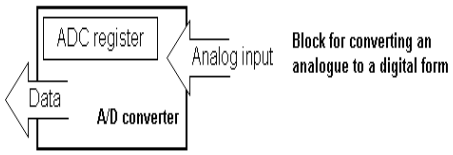


Fig 2 : Analog to Digital Conversion

B. MEMORY ALLOCATION

The Data EEPROM and FLASH Program Memory are readable and writable during normal operation. The data memory is not directly mapped in the register file space. Instead it is indirectly addressed through the Special Function Registers (SFR). There are six SFRs used to read and write the program and data EEPROM memory. The program memory cannot be accessed during the write, therefore code cannot execute. During the write operation, the oscillator continues to clock the peripherals, and therefore they continue to operate when the write completes, the next instruction in the pipeline is executed and the branch to the interrupt vector address will occur. These devices can have up to 8K words of program EEPROM with an address range from 0h to 3FFFh.

The Master Synchronous Serial Port (MSSP) module is a serial interface useful for communicating with other peripheral or microcontroller devices. These peripheral devices may be serial EEPROMs, shift registers, display drivers, A/D converters, etc. The MSSP module can operate in one of two modes:

- Serial Peripheral Interface (SPI)
- Inter-Integrated Circuit (I2C)

C. SERIAL PERHERIAL INTERFACE

The master can initiate the data transfer at any time because it controls the serial clock (SCK). The master determines when the slave is to broadcast data by the software protocol. If the SPI module is only going to receive, the SDO output could be disabled (programmed as an input).The SSPSR register will continue to shift

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 9, September 2013

in the signal present on the SDI pin at the programmed clock rate. As each byte is received, it will be loaded into the SSPBUF register as if a normal received byte (interrupts and status bits appropriately set). The clock polarity is selected by appropriately programming bit CKP (SSPCON<4>).

In slave mode, the data is transmitted and received as the external clock pulses appear on serial clock (SCK). When the last bit is latched, the interrupt flag bit SSPIF (PIR1<3>) is set. While in slave mode, the external clock is supplied by the external clock source on the SCK pin. This external clock must meet the minimum high and low times as specified in the electrical specifications. While in sleep mode, the slave can transmit/receive data. When a byte is received, the device will wake-up from sleep.

III. CAN (CONTROLLER AREA NETWORK) DESCRIPTION

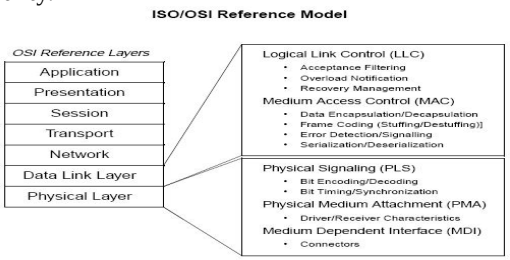
CAN include the Media Dependant Interface definition such that all of the lower two layers are specified.

- ISO11898 is a standard for high-speed applications,
- ISO11519 is a standard for low-speed applications,
- J1939 (from SAE) is targeted for truck and bus applications.

All three of these protocols specify a 5V differential electrical bus as the physical interface. Higher Layer Protocols (HLPs) are generally used to implement the upper five layers of the OSI Reference Model. HLPs are used to:

1) Standardize startup procedures including bit rates used,
2) Distribute addresses among participating nodes or types of messages,
3) Determine the structure of the messages, and
4) Provide system-level error handling routines.

This is by no means a full list of the functions HLPs perform; however it does describe some of their basic functionality.



CAN PROTOCOL UNIT

CAN protocol unit deals with CAN Bus technology through which the Distributed Control System concept is implemented using Embedded PIC Microcontroller. CAN Protocol consist of two layers of OSI layer such as Data link layer, Physical layer with the CAN Controller and Physical layer deals.

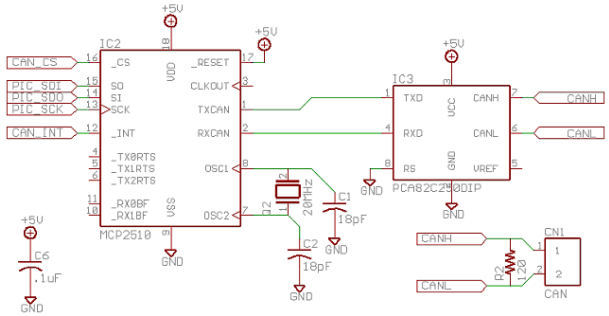


Fig:3 CAN CONTROLLER DESCRIPTION

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 9, September 2013

CAN Controller (MCP2515):

MCP2515 is a Controller Area Network (CAN) protocol controller implementing CAN specification V2.0. It supports CAN 1.2, CAN 2.0A, CAN 2.0B Passive, and CAN 2.0B Active versions of the protocol, and is capable of transmitting and receiving standard and extended messages. It is also capable of both acceptance filtering and message management. It includes three transmit buffers and two receive buffers that reduce the amount of microcontroller (MCU) management required. The MCU communication is implemented via an industry standard Serial Peripheral Interface (SPI) with data rates up to 5 Mb/s.

Device Functionality

The MCP2510 is a stand-alone CAN controller developed to simplify applications that require interfacing with a CAN bus. The device consists of three main blocks:

1) The CAN protocol engine.
2) The control logic and SRAM registers that are used to configure the device and its operation.
3) The SPI protocol block.

The CAN protocol engine handles all functions for receiving and transmitting messages on the bus. Messages are transmitted by first loading the appropriate message buffer and control registers. Transmission is initiated by using control register bits, via the SPI interface or by using the transmit enable pins. Status and errors can be checked by reading the appropriate registers.

Any message detected on the CAN bus is checked for errors and then matched against the user defined filters to see if it should be moved into one of the two receive buffers. The MCU interfaces to the device via the SPI interface. Writing to and reading from all registers is done using standard SPI read and write commands. Interrupt pins are provided to allow greater system flexibility. There is one multi-purpose interrupt pin as well as specific interrupt pins for each of the receive registers that can be used to indicate when a valid message has been received and loaded into one of the receive buffers. There are also three pins available to initiate immediate transmission of a message that has been loaded into one of the three transmit registers. Use of these pins is optional and initiating message transmission can also be done by utilizing control registers accessed via the SPI interface.

CAN Transceiver (MCP 2551):

The MCP 2551 is the interface between the CAN protocol controller and the physical bus. The device provides differential transmit capability to the bus and differential receive capability to the CAN controller. It is main responsible for the convert the differential voltage into normal voltage which is given to bus.

CAN Bus Description

The CAN bus is a balanced (differential) 2-wire interface running over either a Shielded Twisted Pair (STP), Un-shielded Twisted Pair (UTP), or Ribbon cable. The Non Return to Zero (NRZ) bit encoding (with bit-stuffing) is used for data communication on a differential two wire bus. The use of NRZ encoding ensures compact messages with a minimum number of transitions and high resilience to external disturbance.

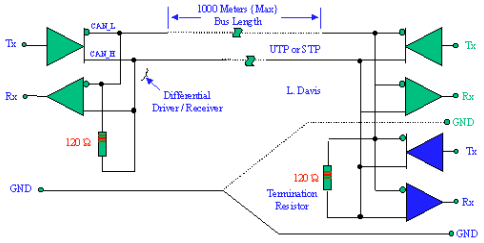


Fig 4 : CAN BUS DESCRIPTION

A number of different data rates are defined, with 1Mbps (Bits per second) being the top end, and 10kbps the minimum rate. The maximum line length is 1Km, 40 meters at 1Mbps. Termination resistors are used at each end of the cable. The worst-case transmission time of an 8-byte frame with an 11-bit identifier is 134 bit times (that's 134 microseconds at the maximum baud rate of 1Mbits/sec). The CAN Bus interface uses an

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 9, September 2013

asynchronous transmission scheme controlled by start and stop bits at the beginning and end of each character. Information is passed from transmitters to receivers in a data frame.

CARRIER SENSE MULTIPLE ACCESS WITH COLLISION AVOIDANCE (CSMA/CA)

The CAN communication protocol is a CSMA/CA protocol. The CSMA stands for Carrier Sense Multiple Access that enable every node on the network must monitor the bus for a period of no activity before trying to send a message on the bus (Carrier Sense). Also, once this period of no activity occurs, every node on the bus has an equal opportunity to transmit a message (Multiple Access). In CAN protocol, the entire arbitration take place without corruption or delay of the higher priority message. There are a couple of things that are required to support non-destructive bitwise arbitration.

- Logic states need to be defined as dominant or recessive.
- The transmitting node must monitor the state of the bus to see if the logic state.

CAN controller define as logic bit 0 as a dominant bit and logic bit 1 as a
recessive bit. A dominant bit state will always in arbitration over a recessive bit state, therefore the lower the value in the Message Identifier (the field used in the message arbitration process), the higher the priority of the message.

CAN Message Frame Description

Standard Data Frame

In common with all other frames, the frame begins with a Start of Frame (SOF) bit, which is of the dominant state, which allows hard synchronization of all nodes.

Arbitration field

The SOF is followed by the arbitration field. Arbitration field consisting of 12 bits; the 11-bit identifier and the Remote Transmission Request (RTR) bit. The RTR bit is used to distinguish data frame (RTR bit dominant) from a remote frame (RTR bit recessive).

Control field

Control field Consisting of six bits; the first bit of this field is the Identifier Extension (IDE) bit which must be dominant to specify a standard frame. The following bit, Reserved Bit Zero (RB0), is reserved and is defined to be a dominant bit by the can protocol. The remaining four bits of the control field are the Data Length Code (DLC) which specifies the number of bytes of data contained in the message.

Data field

This contains any data bytes that are being sent, and is of the length defined by the DLC above (0-8 bytes).

Cyclic Redundancy Check (CRC) Field

It follows the data field and is used to detect transmission errors. The CRC Field consists of a 15-bit CRC sequence, followed by the recessive CRC Delimiter bit. During the ACK Slot bit, the transmitting node sends out a recessive bit. Any node that has received an error free frame acknowledges the correct reception of the frame by sending back a dominant bit. The recessive acknowledge delimiter completes the acknowledge field and may not be overwritten by a dominant bit.

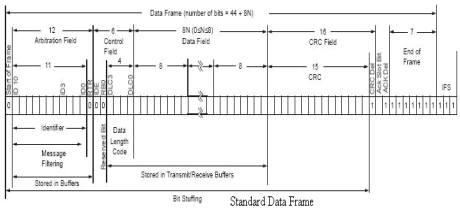


Fig 5: Standard Data Frame

Extended Data Frame

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 9, September 2013

Arbitration field

In the Extended CAN Data Frame, the SOF bit is followed by the arbitration field which consists of 32 bits.The first 11 bits are the most significant bits (Base-lD) of the 29-bit identifier. These 11 bits are followed by the Substitute Remote Request (SRR) bit which is defined to be recessive. The SRR bit is followed by the lDE bit which is recessive to denote an extended CAN frame. It should be noted that if arbitration remains unresolved after transmission of the first 11 bits of the identifier, and one of the nodes involved in the arbitration is sending a standard CAN frame (11-bit identifier), then the standard CAN frame will win arbitration due to the assertion of a dominant lDE bit. Also, the SRR bit in an extended CAN frame must be recessive to allow the assertion of a dominant RTR bit by a node that is sending a standard CAN remote frame. The SRR and lDE bits are followed by the remaining 18 bits of the identifier (Extended lD) and the remote transmission request bit. To enable standard and extended frames to be sent across a shared network, it is necessary to split the 29- bit extended message identifier into 11-bit (most significant) and 18-bit (least significant) sections. This split ensures that the lDE bit can remain at the same bit position in both standard and extended frames.

Control field

Following the arbitration field is the six-bit control field. The first two bits of this field are reserved and must be dominant. The remaining four bits of the control field are the Data Length Code (DLC) which specifies the number of data bytes contained in the message. The remaining portion of the frame (data field, CRC field, acknowledge field, end of frame and intermission) is constructed in the same way as for a standard data.

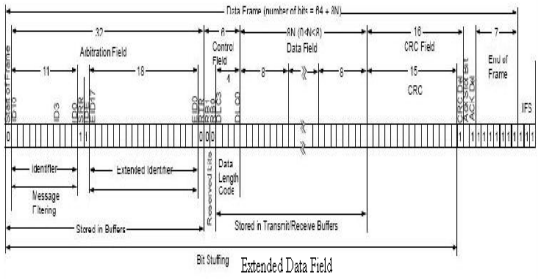


Fig 6: Extended Data Frame

Remote Frame

Normally, data transmission is performed on an autonomous basis by the data source node (e.g. a sensor sending out a data frame). It is possible, however, for a destination node to request data from the source. To accomplish this, the destination node sends a remote frame with an identifier that matches the identifier of the required data frame. The appropriate data source node will then send a data frame in response to the remote frame request.
There are two differences between a remote frame and a data frame.

- The RTR bit is at the recessive state, and
- There is no data field.

In the event of a data frame and a remote frame with the same identifier being transmitted at the same time, the data frame wins arbitration due to the dominant RTR bit following the identifier. In this way, the node that transmitted the remote frame receives the desired data immediately.

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 9, September 2013

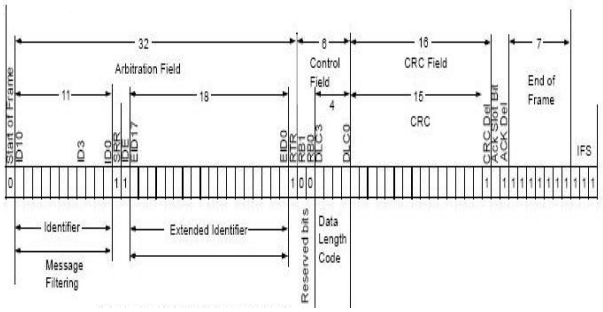


Fig 7: Remote Frame

Error Frame

An Error Frame is generated by any node that detects a bus error.

Error frame consists of two fields,

- An error flag field
- An error delimiter field. There are two types of error flag fields. Which type of error flag field is sent depends upon the error status of the node that detects and generates the error flag field.

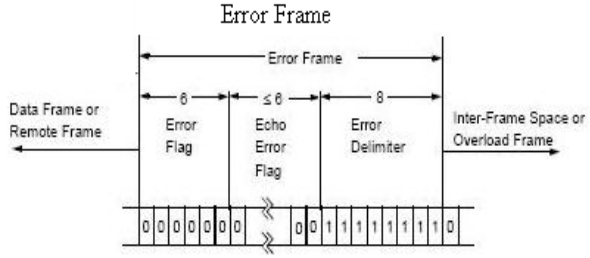


Fig 8: Error Frame

Error-Active Flag

If an error-active node detects a bus error then the node interrupts transmission of the current message by generating an active error flag. The active error flag is composed of six consecutive dominant bits. This bit sequence actively violates the bit stuffing rule. All other stations recognize the resulting bit stuffing error and in turn generate error frames themselves, called error echo flags. The error flag field, therefore, consists of between six and twelve consecutive dominant bits (generated by one or more nodes). The error delimiter field completes the error frame. After completion of the error frame, bus activity returns to normal and the interrupted node attempts to resend the aborted message.

Error-Passive Flag

If an error-passive node detects a bus error then the node transmits an error-passive flag followed by the error delimiter field. The error-passive flag consists of six consecutive recessive bits, and the error frame for an error-passive node consists of 14 recessive bits. From this, it follows that unless the bus error is detected by the node that is actually transmitting, the transmission of an error frame by an error-passive node will not affect any other node on the network. If the transmitting node generates an error-passive flag then this will cause other nodes to generate error frames due to the resulting bit stuffing violation. After transmission of an error frame, an error-passive node must wait for six consecutive recessive bits on the bus before attempting to rejoin bus communications. The error delimiter consists of eight recessive bits and allows the bus nodes to restart bus communications cleanly after an error has occurred.

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 9, September 2013

Overload Frame

An overload frame however can only be generated during an inter frame space. In this way an overload frame can be differentiated from an error frame (an error
frame is sent during the transmission of a message).
The overload frame consists of two fields,

- An overload flag - six dominant bits
- An overload delimiter - eight recessive bits.

An overload frame can be generated by a node as a result of two conditions.

- The node detects a dominant bit during the inter frame space which is an illegal condition.

- Internal conditions the node is not yet able to start reception of the next message.

A node may generate a maximum of two sequential overload frames to delay the start of the next message.

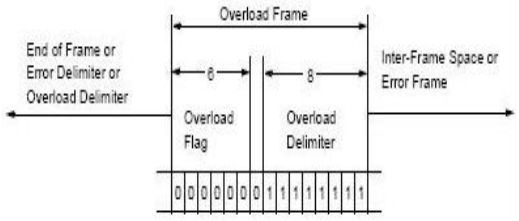


Fig 9:Overload Frame

Errors Detected

CRC Error

A 15-bit Cyclic Redundancy Check (CRC) value is calculatedby the transmitting node and this 15-bit value istransmitted in the CRC field. All nodes on the networkreceive this message calculates a CRC and verifies that the CRC values match. If the values do not match, aCRC error occurs and an Error Frame is generated.Since at least one node did not properly receive themessage, it is then resent after a proper intermissiontime.

Acknowledge Error

The transmittingnode checks if the Acknowledge Slot contains a dominant bit. Thisdominant bit would acknowledge that at least onenode correctly received the message. If this bit isrecessive, then no node received the message properly.An Acknowledge

Form Error

If any node detects a dominant bit in one of the following four segments of the message: End of Frame, Inter frame Space, Acknowledge Delimiter or CRC Delimiter, the CAN protocol defines this to be a form violation and a Form Error is generated. The original message is then resent after a proper intermission time.

Bit Error

A Bit Error occurs if a transmitter sends a dominant bit and detects a recessive bit, or if it sends a recessive bit and detects a dominant bit when monitoring the actual bus level and comparing it to the bit that it has just sent. If a Bit Error is detected, an Error Frame is generated and the original message is resent after a proper intermission time.

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 9, September 2013

Stuff Error
CAN protocol use a Non-Return–to-Zero (NRZ) transmission method this means that the bit level is placed on the bus for the entire bit time. Receiving no dessynchronize on recessive to dominant transitions. If there are more than five bits of the same polarity in a row, CAN will automatically stuff an opposite polarity bit in the data stream.

Error State
Detected errors are made public to all other nodes via Error Frames or Error Flags. Each node is in one of three error states, Error-Active, Error-Passive or Bus-Off.

Error Active
A node is Error-Active when both the Transmit Error Counter (TEC) and the Receive Error Counter (REC) are below 128.Error-Active is the normal operational mode, allowing the node to transmit and receive without restrictions.

IV. CONCLUSION

In this article, the innovation is the flexible use of the CAN bus structure. Connect the CAN bus interface circuit with the original circuitry, and use of ADC which is inbuilt in the PIC microcontroller which is commonly used in common to control multiple motors in synchronization. In the realization of the software input several groups of digital signals, after conversion, output the analog voltage, after a power amplify the signal, control each motor into 12 stalls subdivision. The CAN bus has many advantages such as data reduction, high reliability, real time. These advantages make the system more reliable, closely and stability in the software design.