Contents

[1. Introduction 2](#_Toc73559676)

[2. Design Concept 3](#_Toc73559677)

[3. Protocol Format 4](#_Toc73559678)

[3.1 Destination Address: 4](#_Toc73559679)

[3.2 Source Address: 4](#_Toc73559680)

[3.3 Message Type: 4](#_Toc73559681)

[3.4 Packet Length: 5](#_Toc73559682)

[3.5 Data 1 to N/Payload: 5](#_Toc73559683)

[3.6 CRC: 5](#_Toc73559684)

[4. Message Topology 6](#_Toc73559685)

[5. Challenges 8](#_Toc73559686)

[6. Solution for Challenges 9](#_Toc73559687)

# Introduction

The multi device monitoring system shall accept the different message from network. The network to connect the system can be wired or wireless. The monitoring device shall accept different measurement parameters from sensors, actuators, etc. Each device sends data with unique identification number or by name. The monitoring device shall process data from each device and produces the statistical information like number of messages, number of messages from each device.

# Design Concept

The monitoring device will use RS485, Full Duplex protocol to achieve the goal of multiple device monitoring.

The architectural concept is shown in below diagram:



*Figure1. High Level concept of Device connection*

The monitoring device and other connected devices will use RS485 trans receiver IC.

The monitoring device will use STM based microcontroller and interface between MCU and RS485 trans receiver IC will be based on UART.

# Protocol Format

The protocol between monitoring device and producer device is as shown in below figure:



*Figure 2. High Level concept of Device connection*

## Destination Address:

The monitoring device will be assigned an address as 255.

## Source Address:

The source address of message producer. It indicates from which location message come.

## Message Type:

The message ID will be of 2-byte, higher byte will have message ID for different messages from device. This Message ID decides type of data transfer to monitoring device.

The lower byte will have 5 bits as reserved for future scope.

The 1-bit information of HearBeat indicates that this message is for keep alive information.

The 1-bit information of command indicates that message has been sent from device to monitoring device. This bit will be always set if device wants to pass message to monitoring device.

The 1-bit information of ACK indicates that message has been sent from monitoring device as ACK of receive packet with same message ID and transaction ID. This bit will be always set from monitoring device while transmitting response/ACK packet to device.

## Packet Length:

The packet length will indicate the payload size in message of 8 bit. This will be set as zero when device sends HeatBeat or KeepAlive signal to monitoring device.

## Data 1 to N/Payload:

It indicates actual payload in message. Total number of data will be decided based on Packet Length

## CRC:

16-bit CRC to check integrity of message.

Once monitoring device receives the message, it will perform sanity check with CRC. If calculated CRC matches with received CRC, monitoring device will send ACK packet as shown below

to device with same Destination addres, Message ID and Transaction ID. The Message ID structure remains same as mentioned above. Here Message ID will be change command bit and ACK bit to 0 and 1 respectively.



*Figure3. High Level concept of Device connection*

The Destination Address is the address to which monitoring device is sending response/ACK.

The source address of monitoring Device.

The Message ID structure remains same as mentioned above, Here Message ID will be change command bit and ACK bit to 0 and 1 respectively.

# Message Topology

The monitoring device will use below describe message topology



Always Device initiates communication to monitoring device.

Each device sends the HeartBeat/KeepAlive signal to monitoring device at a rate mentioned below equation:

HeatBeat/KeepAlive rate = (100 \* Source Address) msec.

Monitoring device will monitor HeatBeat/KeepAlive signal from each device. If monitoring device doesn’t receive HeatBeat/KeepAlive signal from device for 700msec, monitoring device will generate an error against that particular device.

Each device will be assigned a source address. Most critical device in bus will have lowest source address. This will be used to determine the priority of message when bus is conflict.

The monitoring device will send ACK packet back to device if message type is command type message. Meaning that device is trying to send some information and monitoring device has received that information.

In above Figure1, when two devices are trying to send the data at the same time, RS485 bus contention will happen. During bus contention data will be corrupted and monitoring device will receive corrupted data. As per message format, monitoring device will first check the sanity of message and pass the message if CRC matches calculated CRC against received CRC. In this scenario CRC will not match hence monitoring device will discard the message.

In above problem, Device has not received the ACK packet from monitoring device in certain time duration(Dead time calculated based on baudrate), Device will assume that bus conflict has happened.

During this time, it will calculate the time based on below formula to send the next message based on Destination Address, whoever has lowest Destination Address will initiate the communication first.

Delay to send next message in bus conflict = 100 + (20 \* Destination Address) msec

For Example:

Destination Address 1 and Destination Address 5 are trying to send the data, None of the device will receive the ACK message from monitoring device hence Destination Address 1 and Destination Address 5 will calculate the delay to send next packet as follow:

Device ID 1 delay = 100 + (20\*1) msec

= 120 msec

Device ID 5 delay = 100 + (20\*5) msec

= 200 msec

# Challenges

RS485 doesn’t support Bus contention concept or Arbitration concept.

Due to this we must consider bus contention or conflict related issues in application layer as explain above using Destination Address concept with example.

# Solution for Challenges

The solution for above mentioned challenge is to have all devices connected over CAN network.

Can does supports bus contention and Arbitration via Message ID.

I have Basic knowledge/understanding on CAN protocol and doesn’t have resources to work on CAN like development board/tools, debugging tool. Hence, I have chosen the similar differential protocol for given problem statement.

By using CAN bus arbitration method, highest priority message will be always received by monitoring device. This is most important in safety critical application.

The solution describes in this document using RS485 will have to handle Bus conflict via application which involves resending the complete message. This might cause problem in mission/time critical application.