Cold Codec Protocol

Introduction

This document describes how the data from the devices can be extracted, understood and the format in which data has to be sent back to the device.

There would also be a list of all data points that devices can generate along with the possible instructions that the server can instruct the hardware to do.

Contents

Overview

Device refers to the hardware module.

Whenever *server* is used it refers to the backend processing done by the Server or the mobile application or any related system.

Packet Overview

The packet is a variable length byte coded packet sent over TCP layer.

This data would be sent to a server with a fixed IP address and port, which can be changed using commands from the server.

The server will have to respond to each packet sent from the device in order to validate the delivery, only after the successful delivery of a packet it will be deleted from the memory of the device. If the device memory gets full the user can lose data. This can happen if the device goes into a low GSM connectivity area for an extended period of time or the server is down.

This is the general packet structure.

Header	Length	Type	Data	CRC
4 Bytes	1 Byte	1 Byte	Variable	4 Bytes

Header

The header of the packet is a 4 byte fixed string which makes it easy to identify the packet. It also greatly reduces the load on the server in some cases when junk data is being sent toward the server, as the application does not need to process the complete packet and can reject it at a very early stage.

67	79	76	68
‘C’	‘O’	‘L’	‘D’

Length

Packet length is a one byte variable which is equal to
TOTAL - (HEADER + PACKET_LENGTH + CRC)

Packet length for a 70 Bytes packet would be
70 - (4+1+4)
70 - 9
61

Type

This is a 1 Byte variable which determines the category of the packet.
Different packet types will have different methods to decode them.

Packet type is divided into 2 nibbles. Upper nibble is for upstream packets and lower nibble is for downstream packets.
Stream direction is decided by processing power. (SERVER > PC/PHONE > DEVICE)

IE: packet from device to pc should use upper nibble.
packet from Server to pc should use lower nibble.

There can be 16 packets in both categories.

Index(XXXX-0000)	Category
0000-0000 | 0	Bidirectional (Unused)
0001-0000 | 16	USB debugging
0010-0000 | 32	Low Priority Standard Packet from device to server
0011-0000 | 48	Medium Priority Standard Packet from device to server
0100-0000 | 64	High Priority Standard Packet from device to server
0101-0000 | 80	Low Priority Extended Packet from device to server
0110-0000 | 96	Medium Priority Extended Packet from device to server
0111-0000 | 112	High Priority Extended Packet from device to server
1000-0000 | 128	
1001-0000 | 144	
1010-0000 | 160	
1011-0000 | 176	
1100-0000 | 192	
1101-0000 | 208	
1110-0000 | 224	
1111-0000 | 240	

Index(0000-XXXX)	Category
0000-0000 | 0	Bidirectional (Unused)
0000-0001 | 1	USB Commands
0000-0010 | 2	Standard Packet from server to device
0000-0011 | 3	Extended Packet from server to device
0000-0100 | 4	Request packet from server to device
0000-0101 | 5	Command packet from server to device
0000-0110 | 6	Acknowledgment packet from server to device
0000-0111 | 7	
0000-1000 | 8	
0000-1001 | 9	
0000-1010 | 10	
0000-1011 | 11	
0000-1100 | 12	
0000-1101 | 13	
0000-1110 | 14	
0000-1111 | 15	

Data

This variable length data is dependent on the packet type and will be discussed later in detail.

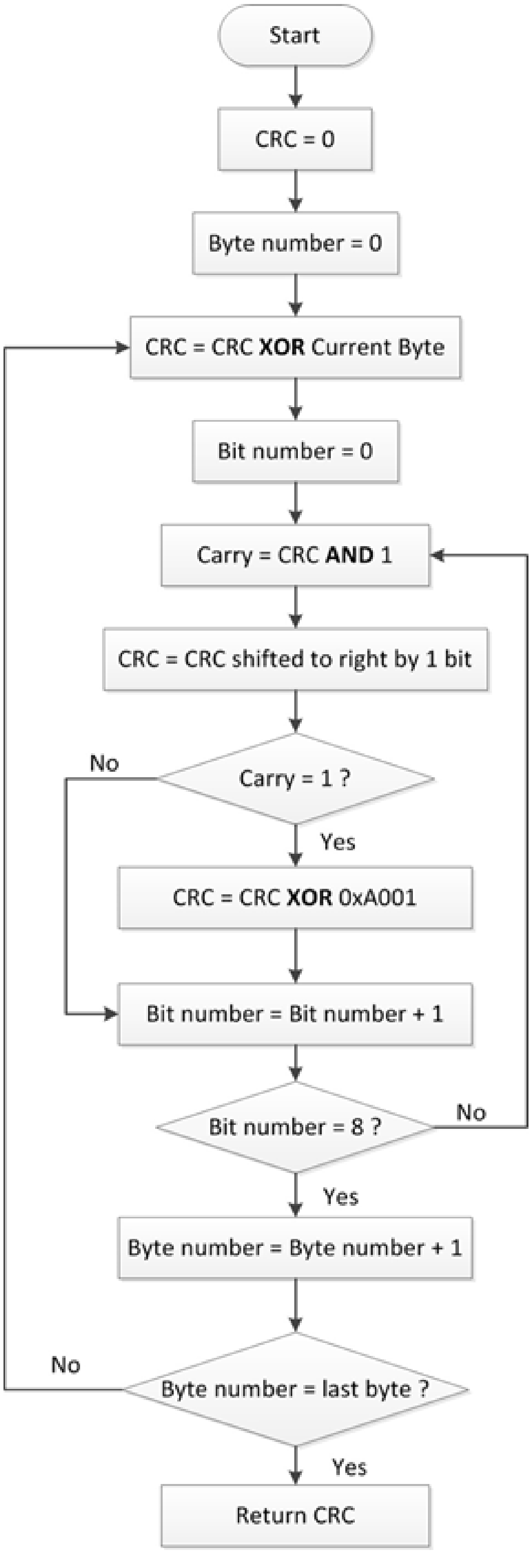
CRC

This is a 4 byte value which is used to determine the validity of the received packet.

The CRC should be calculated for the packet length contents only packet type and data.
If CRC cannot be calculated due to some reason it should be replaced with the HEADER.

48	48	48	48
‘0’	‘0’	‘0’	‘0’

The algorithm to calculate CRC is CRC-16 (also known as CRC-16-IBM).
As this is a CRC-16, for now the first 2 bytes will always be 0.


Start
CRC = 0
Byte number = 0
CRC = CRC **XOR** Current Byte
Bit number = 0
Carry = CRC **AND** 1
CRC = CRC shifted to right by 1 bit
Carry = 1 ?
No
Yes
CRC = CRC **XOR** 0xA001
Bit number = Bit number + 1
Bit number = 8 ?
No
Yes
Byte number = Byte number + 1
Byte number = last byte ?
No
Yes
Return CRC

Example will be added later.

Standard Packet (Type - 32)

Standard Packet structure for device to server communication.

This structure is the same for type 32,48,64,80,96,112.

Header	Length	Type	Generation reason	Data Information	Data	CRC
4 Bytes	1 Byte	1 Byte	1 Byte	5 Bytes	Variable	4 Bytes

Type 32 - Generation Reason

This is a one byte variable data which tells the user why the packet was generated.

Index	Reason
0	Unused (Generate error if this is received)
1	Periodic data
2	Power Connected
3	Power Disconnected
4	Custom data can be added here

Type 32 - Data Information

This variable is a fixed 5 byte data which informs about the total data points present in the packet.

Byte Number	Significance
1	Total number of data points
2	Number of 1 Byte data points
3	Number of 2 Byte data points
4	Number of 4 Byte data points
5	Number of Variable Byte data points

Standard packets will never have any variable data so the packets can be processed accordingly.

Extended packets are supposed to have the variable byte data.

Type 32 - Data ID

Data type is big endian.

ID of 1 byte data
Data byte 1
...
...
ID of 2 byte data
Data byte 1
Data byte 2
...
...
ID of 4 byte data
Data byte 1
Data byte 2
Data byte 3
Data byte 4
...
...
ID of variable byte data
Length of variable byte data
Data byte 1
Data byte 2
...
Data byte n
...
...

Data ID can be of 2 types
Configuration ID’s - (1-63)

Health ID’s (64-255)

ID	Data Meaning	Size (Bytes)	Min Value	Max Value
1	Device ID	2	1	Max
2	Custom data can be added here	4	0	Max

ID	Data Meaning	Size (Bytes)	Min Value	Max Value
64	Timestamp	4		
	Custom values can be added here	2	0	100
		2	0	100
		2	0	100
		2	0	100
		4		
		4		
		1		
		1	0	1
		1	0	100
		1	0	1
		1	0	1
		1	0	1
		4	0	Max
		2	0	Max
		4	0	Max
		1	0	1

Acknowledgment Packet (Type - 6)

This is the packet the server has to send to the device after receiving a packet with successful CRC check.

The server should not receive the same packet number again if the response was received by the device.

Header	Length	Type	Device_ID	Packet Number	CRC
4 Bytes	1 Byte	1 Byte	2 Byte	4 Bytes	4 Bytes

Command Packet (Type - 5)

This packet is used to send commands to the device.

Header	Length	Type	Device_ID	Command Number	Command Value	CRC
4 Bytes	1 Byte	1 Byte	2 Byte	1 Bytes	4 Bytes	4 Bytes

This packet is used to enable or disable fuel dispensing on the truck.
The fuel dispensing is disabled by default.

Server can send any of these commands to the modules at any time to perform the tasks.

Command Number	Significance
1	Custom values can be added here
2	
3	
4	
5	

Revision History

Date	Revision	Changes
21/05/2021	1	Initial release