**FACTOR**

Factor is a term that is used in the transmission of signals between ECUs in CAN protocol. The value that is assigned to a factor will get multiplied with the actual data that’s been sent by the ECU to the bus. In general, factor is used to increase the speed of the signal to reach the destination. If the factor is 1, it means that no changes have been done through the factor in the data that’s been transmitted by the ECU.

Consider the Crash ECU with arbitration ID, 0x01. Here the Crash ECU have to send a data to the Bus as 10 which literally means that a Crash has been detected. In this case, the data that’s been sent by the Crash ECU should have to reach faster to the destination, so that ECUs like Airbag ECU will extract the message from the bus soon. We knew that once a crash has detected, the airbag should get deployed. Thus, the speed of message transmission is directly proportional to the action that’ll be taken by other ECUs regarding the message.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Arbitration ID** | **Message** | **Signal** | **Data** | **Factor** |
| 0x01 | CRASH | Crash detected | 10 | 0.1 |

From the above table it is clear that a data is transmitting to the bus from the Crash ECU in which the factor is given as 0.1 which means that the data 100 will get multiplied to the factor 0.1 which gives the final data as 1. Thus, the data with the factor will reach the destination soon than the data without factor.

**OFFSET**

Offset is a term that is also used in the transmission of signals between ECUs in CAN protocol. The value that is assigned to the offset will get added or subtracted based value with the actual data that’s been sent by the ECU to the bus. In short, Offset is a value, which is used to decrease or increase the number of bytes that’s assigned to a data. If the Offset is 0, it means that no changes have been done through the Offset in the data that’s been transmitted by the ECU.

Consider the Acceleration ECU with arbitration ID, 0x02. Here, the acceleration ECU have to send data from the signal, Vehicle velocity as 100(decimal) to the bus. We knew that this decimal value requires 8 bits I.e. 0110 0100 . In this case, to reduce the number of bits or bytes that is allocated to the data, we use offset.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Arbitration id** | **Message** | **Signal** | **Data** | **Offset** |
| 0x02 | Acceleration | Vehicle Velocity | 10 | 90 |

In this case, the actual data that needs to be sent is 100. But, In the above table data is given as 10 and the offset as 90. It means that the data 10 will be allocated 4bits first and the offset value 90 will get added to the given data, 10 which will give the final output as 100 which is the required output.

In this same message when we change the value of data as 110 and the value of offset as –10, initially the data will be assigned with the respective bytes and then the offset value will get subtracted from the data given. Though, changing the values, the output will be same as 100. The only difference between the previous values and this value is just the minus sign, which does the operation of subtraction instead of addition.

**MINIMUM AND MAXIMUM VALUES**

In CAN protocol Minimum and maximum values are two different fields, in which the range of the message will be given by the user itself. Thus, the data transmitting from the ECU will be sent to the bus only if it fits in the range.

For most cars, the normal operating engine temperature is in a range of 195 to 220 degrees Fahrenheit. The user and the other ECUs should get the message from the Engine temperature ECU if it exceeds the value 220. No other data from this ECU needs to be transmitted to the bus if it does not exceed the value 220.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Arbitration ID** | **Message** | **Signal** | **Data**  **(dec**) | **Minimum valu**e | **Maximum value** |
| 0x03 | Engine Temperature | Temperature | 200 | 220 | 1885E+19 |

From the above table, it is clear that an ECU with the Arbitration Id 0x03 is sending a data 200 where the minimum value is assigned as 220 and the maximum value as 1885E+19 by the user. Here the data does not belong to the range that is defined by the user. So, the message will not be transmitted. Thus, using the minimum and maximum value fields, the unwanted message will get neglected which avoids the data traffic.

**TRANSMITTER AND RECEIVER NODE**

It is a field in the CAN protocol which defines the signal of a message is a transmitter or a receiver node for the particular ECU.

|  |  |  |  |
| --- | --- | --- | --- |
| **Arbitration ID** | **Message** | **Signal** | **Tx/Rx** |
| 0x05 | Airbag Message | Airbag status | Tx |
| 0x06 | Seatbelt Message | Seatbelt status | Rx |
| 0x07 | Engine | Engine Status | Rx |

. From the above table, given 3 messages from the ECUs. I.e. Airbag, Seatbelt, Engine .. Considering the Airbag ECU here, the airbag will get deployed only when the seatbelt is Worn and the Engine is in ON condition. So, the Airbag ECU requires two messages from the Engine and Seatbelt ECU. Once, when the data is received from the ECU, the Airbag will either deploy or not deploy based on the received message and the status about the airbag will be sent to the bus. So, in this case, the airbag message acts as a transmitting node and the other two messages acts as a receiving node.