

CRC_Calc (FUN)

Функция для вычисления CRC по протоколу Тензо-М.

- При отправке данных: после блока данных <Data> в буфере должен быть нулевой байт, он используется функцией для вычисления CRC.
- При приёме данных: при проверке в функцию передаются блоки <Data> и <CRC>, в этом случае вычисленный функцией CRC должен быть равным 0.
- Служебные байты 16#FE не должны участвовать в расчёте CRC.

На вход функции необходимо передать указатель на стартовый адрес и размер данных, для которых необходимо вычислить CRC. На выход передаётся вычисленный CRC.

Scope	Name	Type	Comment
Return	CRC_Calc	BYTE	
Input	InData	POINTER TO BYTE	Указатель на буфер данных
	DataLen	INT	Размер данных для расчёта CRC

Объявление переменных

(*

Функция для вычисления CRC по протоколу Тензо-М.

- При отправке данных: после блока данных <Data> в буфере должен быть нулевой байт, он используется функцией для вычисления CRC.

- При приёме данных: при проверке в функцию передаются блоки <Data> и <CRC>, в этом случае вычисленный функцией CRC должен быть равным 0.

- Служебные байты 16#FE не должны участвовать в расчёте CRC.

*)

FUNCTION CRC_Calc: BYTE

VAR_INPUT

InData: POINTER TO BYTE; // Указатель на буфер данных

DataLen: INT; // Размер данных для расчёта CRC

END_VAR

VAR

i, j: INT;

DataBuffer: BYTE; // Внутренний буфер данных

END_VAR

VAR CONSTANT

Polynom: BYTE := 16#69; // Полином для расчёта CRC

END_VAR

Код исполнения

FOR i := 0 TO DataLen DO

DataBuffer := InData[i];

FOR j := 0 TO 7 DO

IF CRC_Calc.7 THEN

CRC_Calc := CRC_Calc * 2;

IF DataBuffer.7 THEN

CRC_Calc := CRC_Calc + 1;

END_IF

CRC_Calc := CRC_Calc XOR Polynom;

ELSE

```
CRC_Calc := CRC_Calc * 2;  
IF DataBuffer.7 THEN  
    CRC_Calc := CRC_Calc + 1;  
END_IF  
END_IF  
DataBuffer := DataBuffer * 2;  
END_FOR  
END_FOR
```

=====