Функция, вычисляющая контрольную сумму для протокола обмена «Тензо-М»

```
//~~~~~~~~~~~~~~~~~~~~~~~~~~
// Эта функция рассчитывает контрольную сумму
// последовательности байтов и возвращает результат.
unsigned char ucCrcMaker
    (unsigned char *InputData,
     unsigned char BytesNumber,
                          unsigned char Offset)
//   *InputData - указатель на последовательность байтов.
//   BytesNumber - количество байтов,
// для которых считается контрольная сумма.
//   Offset - смещение относительно начала последовательности
// байтов, с которого начинает считаться контрольная сумма
// (0 - без смещения).
//   Возвращает контрольную сумму.
{
   // Служебные параметры.
register unsigned char i, j, Data, CrcCode = 0, Polinom = 0x69;

for(i = Offset; i < BytesNumber + Offset; i++)
{
 Data = InputData[i];
 for(j = 0; j < 8; j++)
  {
   if(CrcCode & (1 << 7))
   {
    CrcCode *= 2;
    if(Data & (1 << 7)) CrcCode ++;
    CrcCode ^= Polinom;
   }
   else // if(CrcCode & (1 << 7))
   {
    CrcCode *= 2;
    if(Data & (1 << 7)) CrcCode ++;
   }    // if(CrcCode & (1 << 7))
   Data *= 2;
  } // for(j = 0; j < 8; j++)
}  // for(i = Offset; i < BytesNumber + Offset; i++)

// Вернём контрольную сумму.
return CrcCode;

}  // unsigned char ucCrcMaker ( ... )
//~~~~~~~~~~~~~~~~~~~~~~~~~~
```

Пример применения функции расчёта контрольной суммы при получении сообщения

```
// Буфер протокола на приём сообщений.
// В этом буфере уже не должно быть маркеров байт-стаффинга  (0xFE) и
// маркеров начала и конца сообщения (0xFF).
// Сообщение должно располагаться в буфере с начала буфера (с индекса «0»).
// В приведённом примере размер сообщения в байтах известен до применения функции
// unsigned char ucCrcMaker
                        (unsigned char *InputData, unsigned char BytesNumber,
                        unsigned char Offset);
// В приведённом примере он записан в переменную «ucProtocolTensoMMessageLength»
unsigned char ucProtocolTensoMReceivBuffer[PROTOCOL_TENSO_M_RECEIV_BUFFER_SIZE];

if(
  ucCrcMaker( ( (unsigned char*)(&(ucProtocolTensoMReceivBuffer [0])) ),
                ucProtocolTensoMMessageLength,
              0)
  )

{ // Контрольная сумма НЕ совпала.


} // if(контрольная сумма не совпала)

else

{ // Контрольная сумма совпала.


}
```

Пример применения функции расчёта контрольной суммы при формировании сообщения для отправки

```
// Буфер протокола на отправку сообщений.
unsigned char volatile ucProtocolTensoMTransmitBuffer
                        [PROTOCOL_TENSO_M_TRANSMIT_BUFFER_SIZE];
// ВНИМАНИЕ!!! Буфер заполняется снизу вверх.

// Хвост буфера протокола на передачу (по этому адресу закладывается
// очередной символ для передачи).
unsigned char volatile ucProtocolTensoMTransmitBufferTail;

// Перед отправкой сформированная посылка находится в буфере, начиная с индекса «0».
// Байтов в посылке – «ucProtocolTensoMTransmitBufferTail», маркеров байт-стаффинга  (0xFE) и
// маркеров начала и конца сообщения (0xFF) ещё нет.


// Сначала обнуляем контрольную сумму.
  ucProtocolTensoMTransmitBuffer
   [ucProtocolTensoMTransmitBufferTail] =
    0x00;
    // Теперь рассчитываем реальную контрольную сумму.
  ucProtocolTensoMTransmitBuffer
    [ucProtocolTensoMTransmitBufferTail] =
  ucCrcMaker(((unsigned char*)(&( ucProtocolTensoMTransmitBuffer [0] ))),
      ( ucProtocolTensoMTransmitBufferTail + 1 ),
                    0);


                            // Посылка сообщения.


// Посылаем маркер начала сообщения (0xFF).


// Посылаем по очереди все байты сообщения (включая контрольную сумму) из буфера
// ucProtocolTensoMTransmitBuffer.
// Если очередной посылаемый байт (включая контрольную сумму) равен «0xFF»,
// то после такого байта вставляем байт «0xFE».


// Посылаем два маркера конца сообщения (0xFF).
```