

Артамонов И.И. Метод обнаружения ошибок в ходе передачи данных между электронными устройствами // Академия педагогических идей «Новация». Серия: Студенческий научный вестник. – 2018. – №11 (ноябрь). – АРТ 523-эл. – 0,2 п.л. - URL: <http://akademnova.ru/page/875550>

РУБРИКА: ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

УДК 004.6: 004.7

Артамонов Илья Игоревич
студент 2-го курса магистратуры, факультет ИВТ
Научный руководитель: Айман Аль Аккад, к.т.н.
ФГБОУ ВПО «Ижевский государственный технический университет
имени М. Т. Калашникова»
г. Ижевск, Российская Федерация
email: fernando94@yandex.ru

**МЕТОД ОБНАРУЖЕНИЯ ОШИБОК В ХОДЕ ПЕРЕДАЧИ
ДАННЫХ МЕЖДУ ЭЛЕКТРОННЫМИ УСТРОЙСТВАМИ**

Аннотация: В данной статье происходит обзор одного из методов поиска и предотвращения ошибок в процессе передачи данных между различными электронными устройствами. В статье приведены формулы и таблицы для расчета контрольной суммы, которая помогает избежать ошибок в процессе обмена данными. Также указан алгоритм работы данного метода.

Ключевые слова: контрольная сумма, полином, исключающее ИЛИ, байт, бит.

Artamonov Iliа Igorivich
2nd year master degree student, features of information technologies
Supervisor: Aiman Al Akkad, Candidate of Technical Sciences
FGBOU VPO “Kalashnikov Izhevsk State Technical University”
Izhevsk, Russian Federation

METHOD OF DETECTING ERRORS IN THE TRANSFER OF DATA BETWEEN ELECTRONIC DEVICES

Abstract: In this article, one of the methods of searching and preventing errors in the process of data transfer between various electronic devices is reviewed. The article contains formulas and tables for calculating the checksum, which helps to avoid mistakes in the data exchange process. The algorithm of this method is also indicated.

Key words: checksum, polynomial, exclusive OR, byte, bit.

Шум и кратковременные электрические помехи могут привести к изменению данных по мере их прохождения через канал связи. Если получателю не удастся обнаружить это, полученное сообщение будет неверным, что может привести к серьезным последствиям [1]. В исследовательской работе в процессе передачи данными между контроллером заряда и программой происходит проверка правильной передачи данных. Данная проверка осуществляется алгоритмом под названием «Циклический избыточный код». В этом методе пакеты, составляющие сообщение, добавляются арифметически. Номер контрольной суммы добавляется к последовательности пакетов так, чтобы сумма данных плюс контрольная сумма была равна нулю. Полученная

последовательность пакетов может быть добавлена вместе с контрольной суммой локальным микропроцессором. Если сумма не равна нулю, произошла ошибка. Пока сумма равна нулю, маловероятно (но не невозможно), что какие-либо данные были повреждены во время передачи [2]. Есть две реализации выполнения данного алгоритма для обработки ошибок, первый это простой метод, второй табличный. В данной работе используется табличный метод проверки ошибок. Для каждой версии передачи данных, n-битной передачи данных, за основу берется полином соответствующей степени. В данной работе используется полином 16-й степени и этот полином выглядит следующим образом [3,4]:

$x^{16} + x^{15} + x^2 + 1$ – полином для передачи 16-битных сообщений, где x – число, записанное в виде двоичной строк, биты которого служат коэффициентами полинома. $x \in A, A = \{a_1 + \dots + a_{16}\}, a_i \in B = \{0,1\}$

Контрольная сумма CRC формируется следующим уравнением

$$M(x) * x^n = Q(x) * G(x) + R(x),$$

где $M(x)$ – исходный полином сообщения;

$G(x)$ – многочлен генератора степени;

$M(x) * x^n$ – исходное сообщение с добавленными n нулями в конце сообщения;

$R(x)$ – коэффициент остаточного многочлена, степень которого меньше n ;

$Q(x)$ – фактор полином [5].

Матрица A_{ij} – матрица старших и младших битов, где в первом столбце отображаются старшие байты, во втором младшие биты, данная матрица имеет вид:

$$\begin{pmatrix} a_{0,0} & a_{0,1} \\ \dots & \dots \\ a_{127,0} & a_{127,1} \end{pmatrix}$$

Ниже, на рисунках 1 и 2, представлены таблицы для обработки контрольной суммы для CRC-16 [6]

Таблица старших битов

```

{
  0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81,
  0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0,
  0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01,
  0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,
  0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80,
  0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0,
  0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01,
  0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
  0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81,
  0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1,
  0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
  0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0,
  0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,
  0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81,
  0x40
}
  
```

Рис.1. Таблица старших битов

Таблица младших битов

```
{
0x00, 0xC0, 0x01, 0x01, 0xC3, 0x03, 0x02, 0xC2, 0x06, 0x07, 0xC7, 0x05, 0xC5, 0xC4,
0x04, 0xCC, 0x0C, 0x0D, 0xCD, 0x0F, 0xCF, 0x0E, 0x0A, 0xCA, 0xCB, 0x0B, 0xC9, 0x09,
0x08, 0xC8, 0x08, 0x18, 0x19, 0xD9, 0x1B, 0xDB, 0xDA, 0x1A, 0x1E, 0xDE, 0xDF, 0x1F, 0xDD,
0x1D, 0x1C, 0xDC, 0x14, 0xD4, 0xD5, 0x15, 0xD7, 0x17, 0x16, 0xD6, 0xD2, 0x12, 0x13, 0xD3,
0x11, 0xD1, 0xD0, 0x10, 0xF0, 0x30, 0x31, 0xF1, 0x33, 0xF3, 0xF2, 0x32, 0x36, 0xF6, 0xF7,
0x37, 0xF5, 0x35, 0x34, 0xF4, 0x3C, 0xFC, 0xFD, 0x3D, 0xFF, 0x3F, 0x3E, 0xFE, 0xFA, 0x3A,
0x3B, 0xFB, 0x39, 0xF9, 0xF8, 0x38, 0x28, 0xE8, 0xE9, 0x29, 0xEB, 0x2B, 0x2A, 0xEA, 0xEE,
0x2E, 0x2F, 0xEF, 0x2D, 0xED, 0xEC, 0x2C, 0xE4, 0x24, 0x25, 0xE5, 0x27, 0xE7, 0xE6, 0x26,
0x22, 0xE2, 0xE3, 0x23, 0xE1, 0x21, 0x20, 0xE0, 0xA0, 0x60, 0x61, 0xA1, 0x63, 0xA3, 0xA2,
0x62, 0x66, 0xA6, 0xA7, 0x67, 0xA5, 0x65, 0x64, 0xA4, 0x6C, 0xAC, 0xAD, 0x6D, 0xAF, 0xBF,
0x6E, 0xAE, 0xA8, 0x6A, 0x6B, 0xAB, 0x69, 0xA9, 0xA8, 0x68, 0x78, 0xB8, 0xB9, 0x79, 0xBB,
0x7B, 0x7A, 0xBA, 0xBE, 0x7E, 0x7F, 0xBF, 0x7D, 0xBD, 0xBC, 0x7C, 0xB4, 0x74, 0x75, 0xB5,
0x77, 0xB7, 0xB6, 0x76, 0x72, 0xB2, 0xB3, 0x73, 0xB1, 0x71, 0x70, 0xB0, 0x50, 0x90, 0x91,
0x51, 0x93, 0x53, 0x52, 0x92, 0x96, 0x56, 0x57, 0x97, 0x55, 0x95, 0x94, 0x54, 0x9C, 0x5C,
0x5D, 0x9D, 0x5F, 0x9F, 0x9E, 0x5E, 0x5A, 0x9A, 0x9B, 0x5B, 0x99, 0x59, 0x58, 0x98, 0x88,
0x48, 0x49, 0x89, 0x4B, 0x8B, 0x8A, 0x4A, 0x4E, 0x8E, 0x8F, 0x4F, 0x8D, 0x4D, 0x4C, 0x8C,
0x44, 0x84, 0x85, 0x45, 0x87, 0x47, 0x46, 0x86, 0x82, 0x42, 0x43, 0x83, 0x41, 0x81, 0x80,
0x40
};
```

Рис.2. Таблица младших битов

Алгоритм табличного метода расчета контрольной суммы:

- 1) $b_i \text{ shl } 1 = s$, где s новый байт, прочитанный из поступившего сообщения, а $\text{shl } 1$ -сдвиг регистра на 1 байт влево;
- 2) $b_i \text{ XOR } b_{i+1} = z$ - происходит операция исключающее XOR над старшим байтом, который был только что выдвинут из регистра – b_i , со следующим байтом сообщения – b_{i+1} , что даст индекс в таблице $[0..255] - z$;
- 3) $b_{i+1} \text{ XOR } a_{zk} = c$, где происходит операция XOR над a_{zk} - значения ячейки таблицы с содержанием регистра b_{i+1} , c – значение контрольной суммы;
- 4) если есть еще необработанные байты информации, то перейти к п.1.[7,8]

Заключение. Мы просмотрели один из возможных методов предотвращения ошибок в процессе коммуникации электронных устройств. Изучили алгоритм, благодаря которому происходит расчет контрольной суммы для выявления ошибок. Данный метод будет

применяться для расчета ошибок в процессе обмена информацией между программой и контроллером заряда электропитания.

Список использованной литературы:

- 1) Белоус А.И., Солодуха В.А, Шведов С.В. Основы конструирования высокоскоростных электронных устройств. - Москва, Техносфера, 2017.-872 стр.
- 2) Jessee Russell, Ronald Cohn , Протокол передачи данных – Санкт-Петербург, Книга по требованию, 2015 – 663 стр.
- 3) Verhouz A, Forouzan, Data Communications and Networking – McGraw-Hill, New-York, 2007 – 1171 стр.
- 4) Willam Stallings, Data and computer communication – Pearson Education, London, 2007-901 стр.
- 5) Протоколы передачи данных [Электронный ресурс] URL: http://inf.e-alekseev.ru/text/Protocols_per_dan.html
- 6) Modbus RTU [Электронный документ] URL: <https://ipc2u.ru/articles/prostye-resheniya/modbus-rtu/>
- 7) Hart протокол [Электронный ресурс] URL: http://www.bookasutp.ru/Chapter2_5.aspx
- 8) UART протокол [Электронный документ] URL: http://www.rotr.info/electronics/mcu/arm_usart.htm

Дата поступления в редакцию: 08.11.2018 г.

Опубликовано: 15.11.2018 г.

© Академия педагогических идей «Новация». Серия «Студенческий научный вестник», электронный журнал, 2018

© Артамонов И.И., 2018