Click here for 8051 Microcontroller FORUM

www.8051projects.info

8051 Forum

FREE MICROCONTROLLER PROJECTS

BOOKMARK

÷**Home**
÷**8051 Forum**
÷**Active Topics**
÷**8051 Free Projects**
÷**Blog**
÷**Top 10**
÷**Datasheets**
÷**Electronic Tutorials**
÷**Electronic Projects**
÷**8051 Tutorials**
÷**8051 Course**
÷**8051 Code Library**
÷**Visual Basic for ν C**
÷**8051 Programmer**
÷**Assembler/IDE**
÷**Products**
÷**Ebooks**
÷**Supporting Tools**
÷**Tag Cloud**
÷**Guest Book**
÷**Contact Me**
÷**Privacy policy**

8051 Microcontroller Projects, free ebooks, project codes, AVR Projects, 8051 tutorials, electronic projects, ECE CSE BE final year projects, Embedded systems, AVR codes, embedded technology, LCD Interfacing tutorial,final year project

FBUS communication
………

Here you'll find a description of the protocol used in Nokia phones. Please note that Nokia doesn't provide any official information about the protocol it uses. The source of this information is mainly Gnokii and the hex command heaps that I made by sniffing the communication between the P.C. and Nokia phones. Nokia phones use two kinds of protocols, the Mbus protocol and the Fbus protocol. There is a lot of difference between the two, the basic difference being that one cannot transmit and receive data simultaneously with mbus, but the same is possible using fbus. Also there is a difference in the speeds. While Mbus operates at 9600 bps, Fbus operates at 115200 bps. So, if you want some fast data transfer between the pc and your phone, better use Fbus. Both mbus and fbus documentation can also be found at Gnokii, which is the original source of this document.

Protocol Exchange Layout
1. The Microcontroller sends some type of request or command packet
2. Phone sends acknowledgement
3. Phone sends response to the request packet
4. Microcontroller sends acknowledgement

Protocol Speed
Speed   :: 115,200 bps
Bits   :: 8
Parity   :: None
Stop Bits :: 1

Format of packets from Microcontroller (General packet format)
{ FrameID, DestDEV, SrcDEV, MsgType, 0x00, FrameLength, {block}, FramesToGo, SeqNo, PaddingByte?, ChkSum1, ChkSum2 }

Where **FrameID**   = 0X1c   --- IR/Fbus
                        = 0x1e   --- Serial/Fbus

DestDEV, SrcDEV = 0X00   --- Mobile Phone
                        = 0x0c   --- Microcontroller

MsgTpye   = Indicates the type of message .. can be 0x00 or 0x40 .. I haven't been able to decode it :(