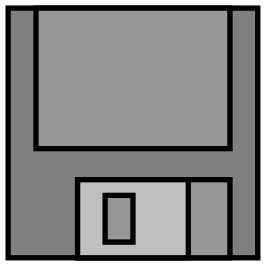
Source Code for
Vending Machine Coin Changer
Audit data Collection

Example Source Code For

DEX/UCS
Audit Data
Retrieval



DEX Audit Information

Introduction

This publication is intended to provide programmers with an interest in DEX audit data collection, with a working example of a DEX audit application, from which they can develop there own audit applications.

Scope

It is assumed that the reader of this publication will have access to the DEX/UCS specification, or at least the EVA DTS specification. Since the example code is written in C/C++, a proficiency in these languages is also required.

DEX Audit Information

The “dexread.exe” example application

On the accompanying disk you will find both an executable copy and the source code for the Dexread application. When running Dexread, you can issue a command line arguments, to specify which communication port you wish to use for DEX communication. If you omit the command line argument, Dexread will default to COM 1

Compiling the program.

On the accompanying disk you will find the following files.

dexread.ide	-	Borlands C++ project file
dex_audit.cpp	-	Main functions for Audit collection
includes.h	-	all necessary includes for main functions
pc_io.cpp	-	I/O routines for DOS operating systems
pc_io.h	-	Includes for IO routines.

These are the files that were used to build the Dexread application.
Notice that this is a DOS based application, so if you are not using the Borlands project file, you should set your compilers target operating system to DOS.

About the Source Code

The example source code was written and compiled using Borlands C++ v5.02. It has been tested under Windows 95 operating system and works as expected.

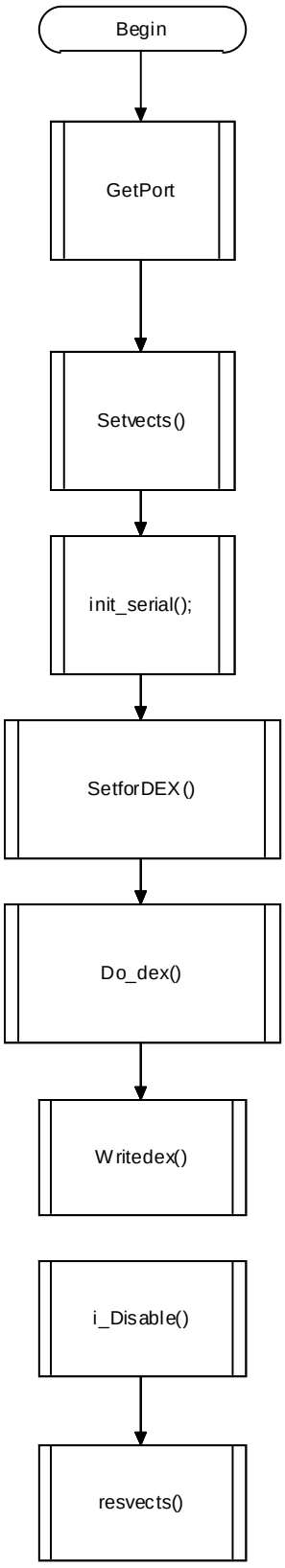
The source code provided can be used under the terms and conditions contained within the disclaimer, however, it is strongly recommended that the reader optimize the code for his/her specific needs, e.g. port it to Windows.

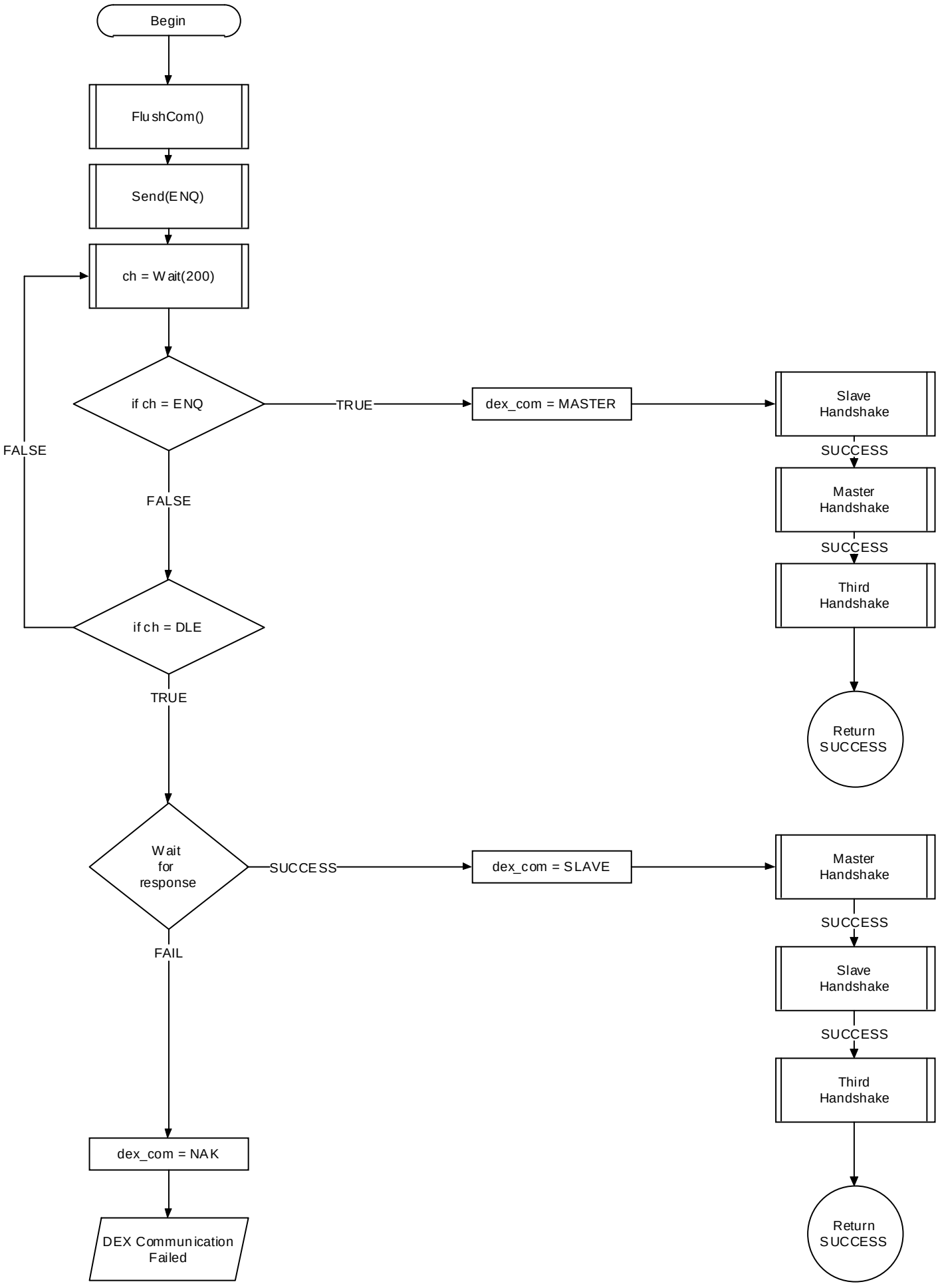
About the Flow Charts

The flow charts have been produced to show the flow of the Dexread program. For simplicity, the variable names used in the flowcharts have been kept the same as those used in the functions supplied in the example source code. Also the program flow has been broken down into separate flow charts representing the separate functions in the program.

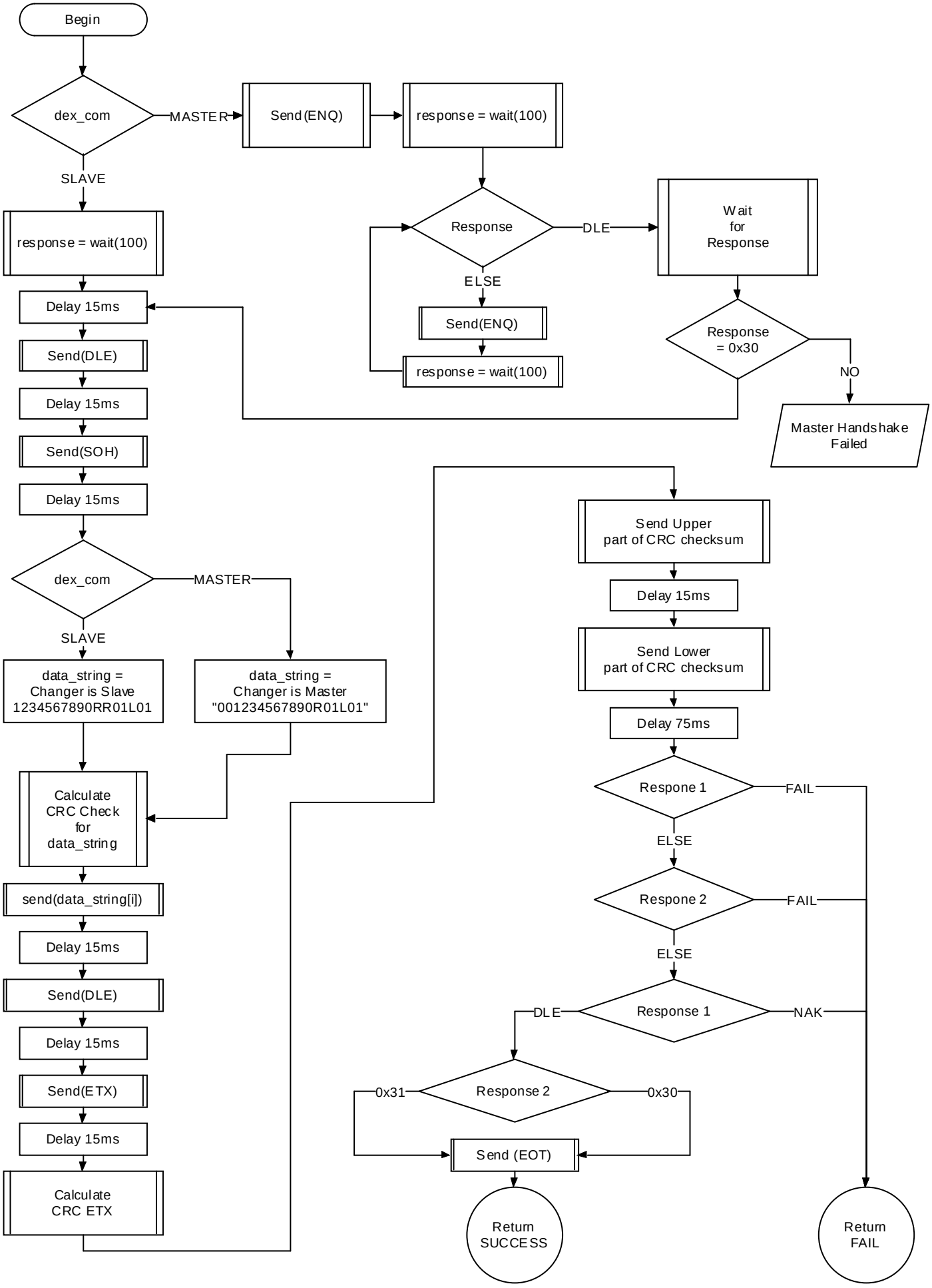
Flow charts for the IO routines have not bee included since these will vary depending on the target platfor and operating system, and it was felt that the inclusion of these could draw the focus away from the DEX routines.

Main Program Function

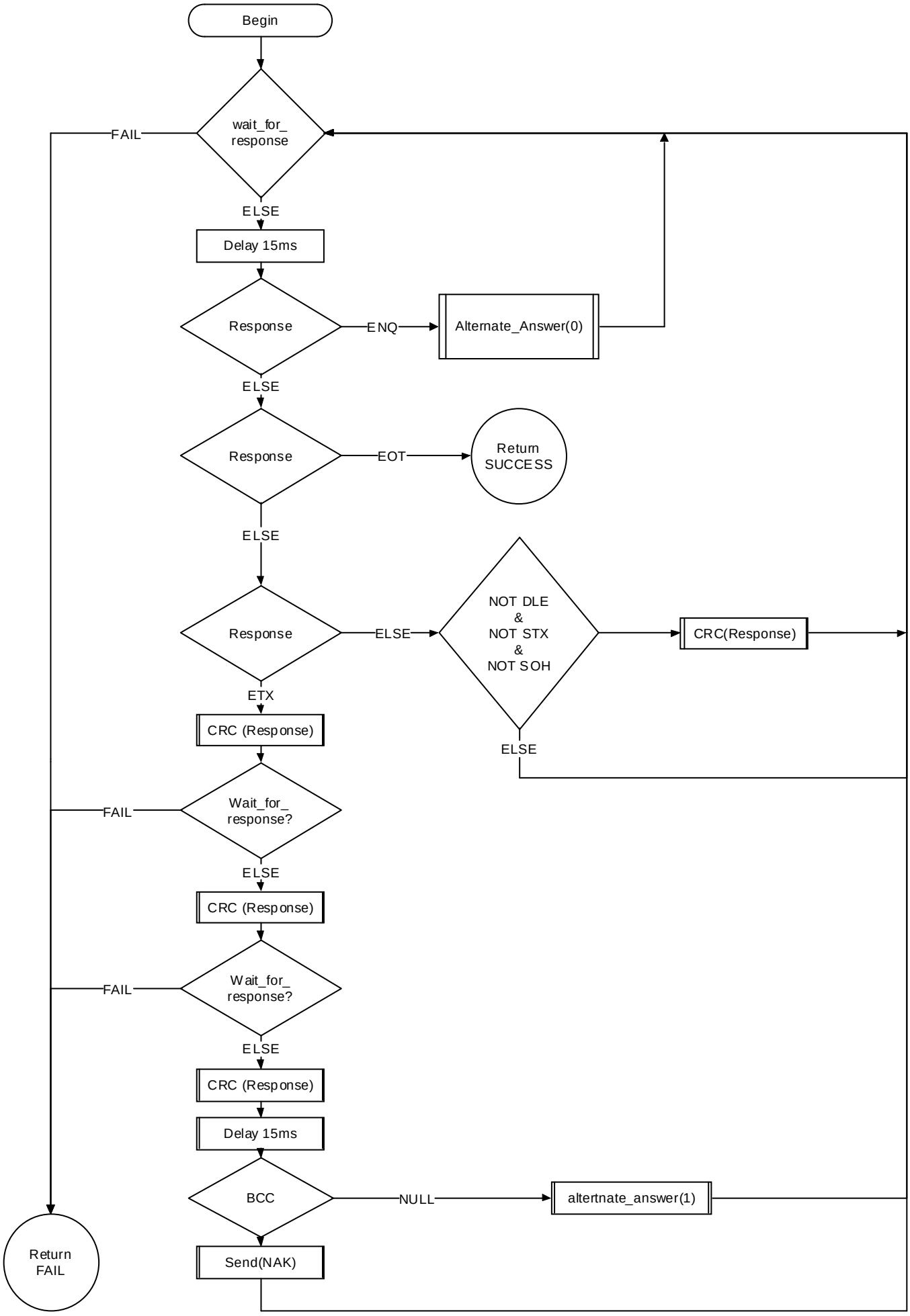



Begin
FlushCom()
Send(ENQ)
ch = Wait(200)
if ch = ENQ
TRUE
dex_com = MASTER
Slave Handshake
SUCCESS
Master Handshake
SUCCESS
Third Handshake
Return SUCCESS
FALSE
FALSE
if ch = DLE
TRUE
Wait for response
SUCCESS
dex_com = SLAVE
Master Handshake
SUCCESS
Slave Handshake
SUCCESS
Third Handshake
Return SUCCESS
FAIL
dex_com = NAK
DEX Communication Failed

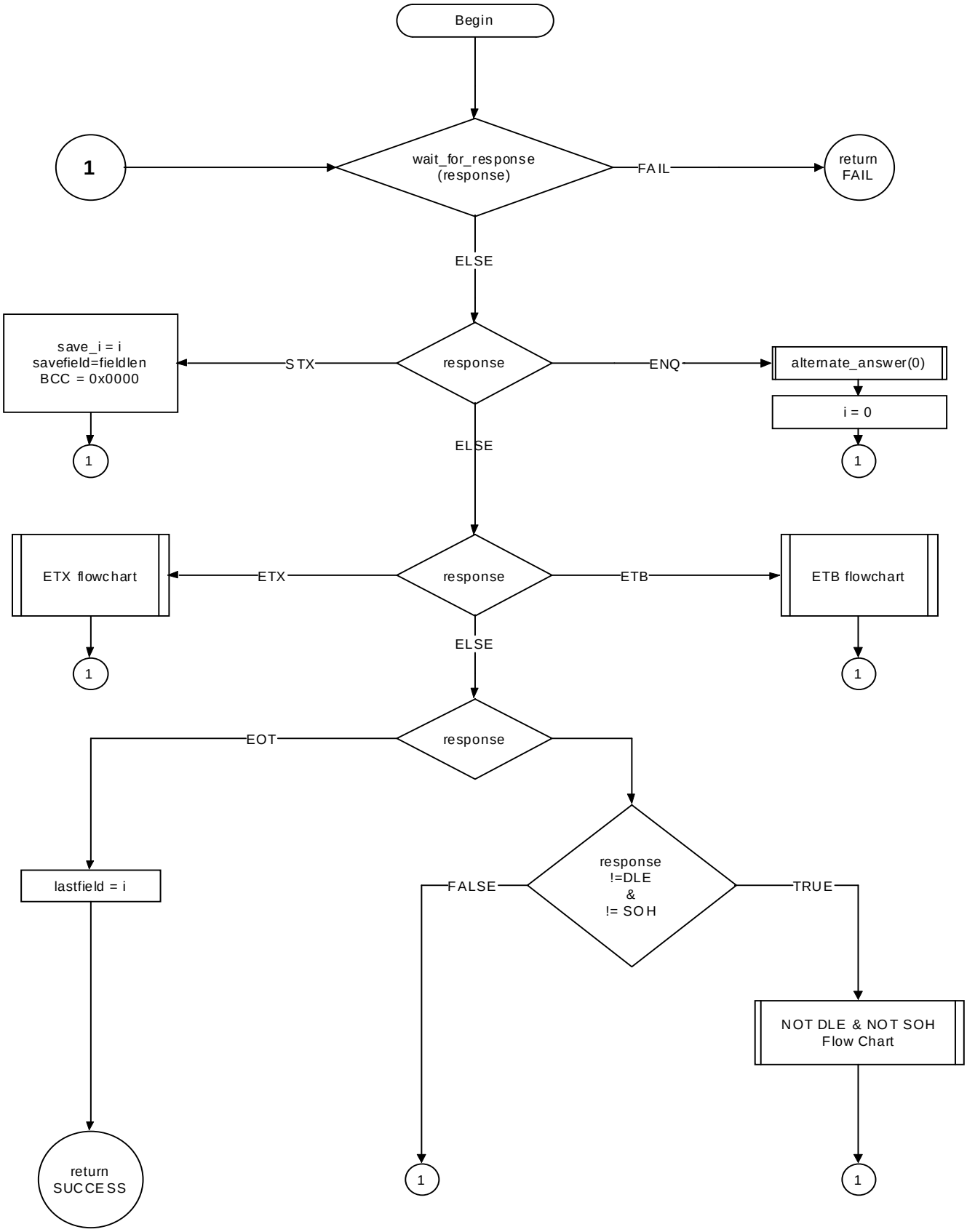
Master Handshake



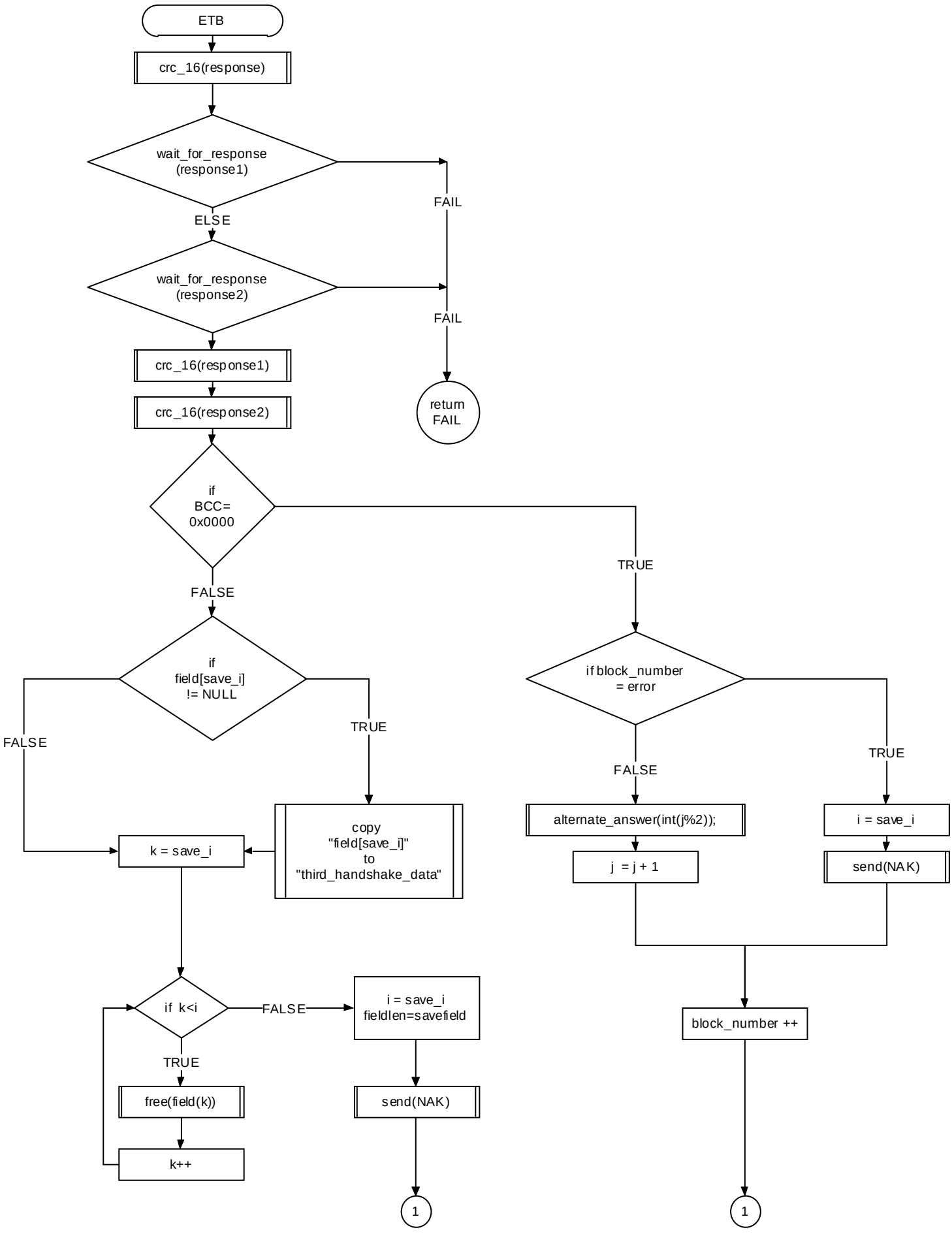
Slave Handshake



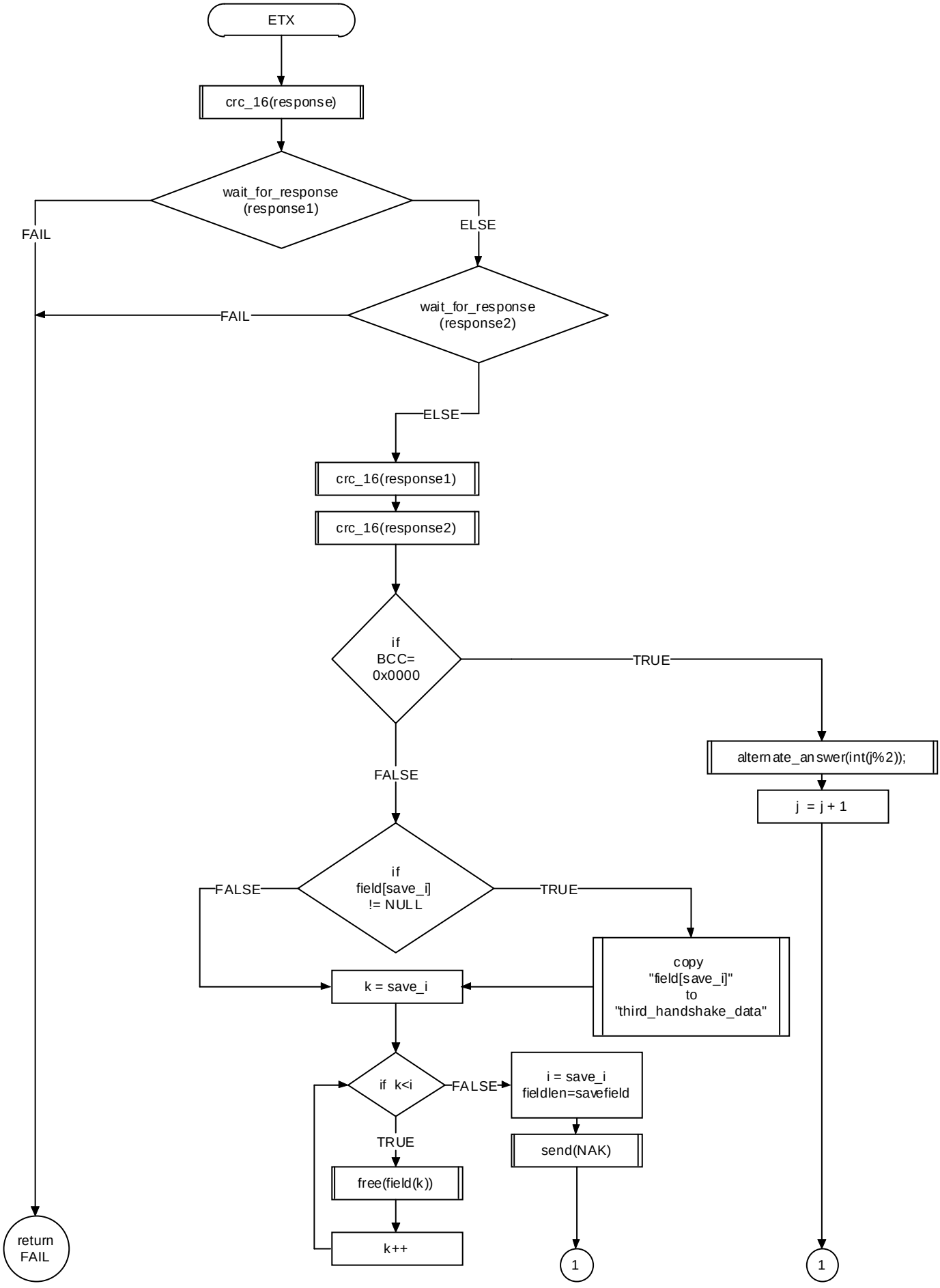
Third Handshake



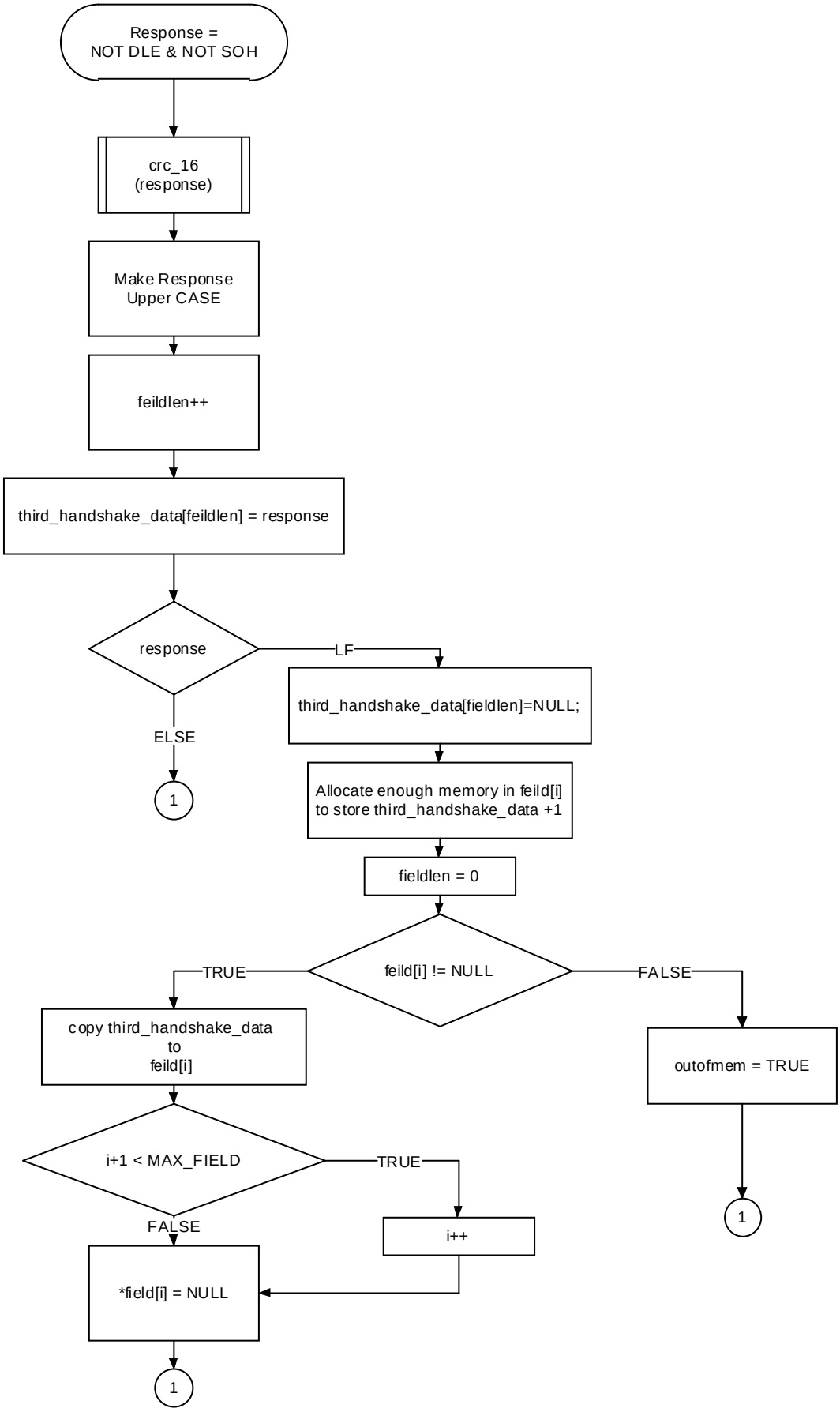
Third Handshake - ETB



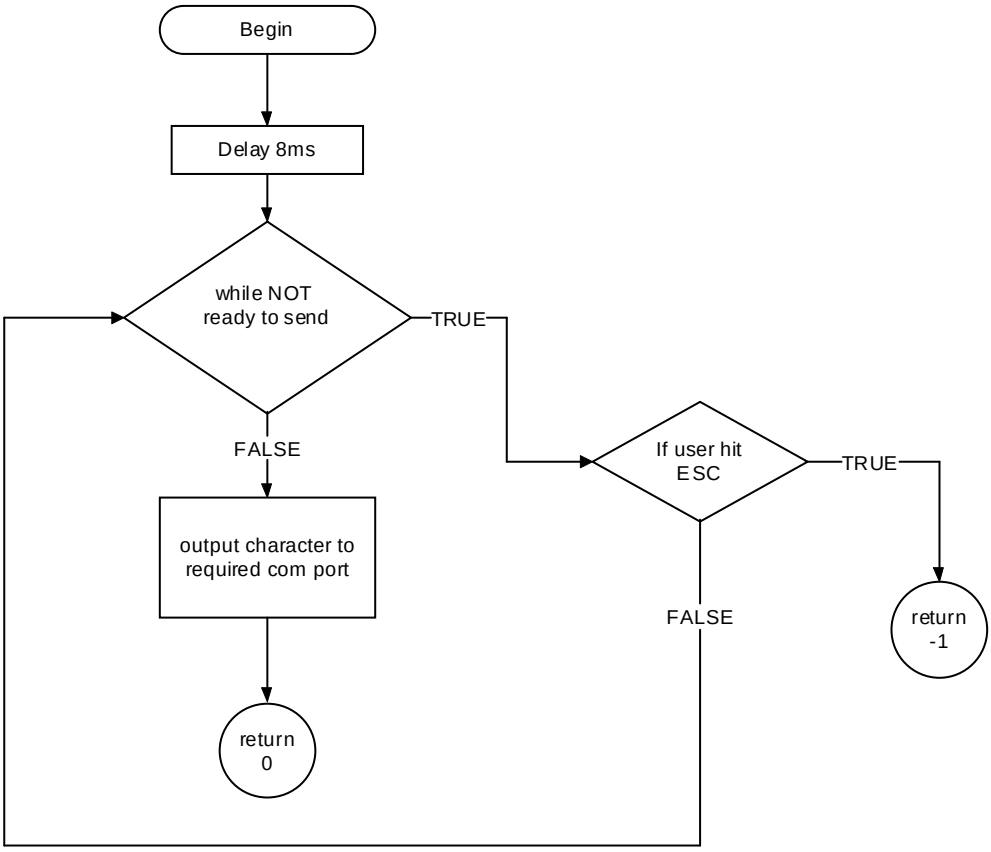
Third Handshake - ETX



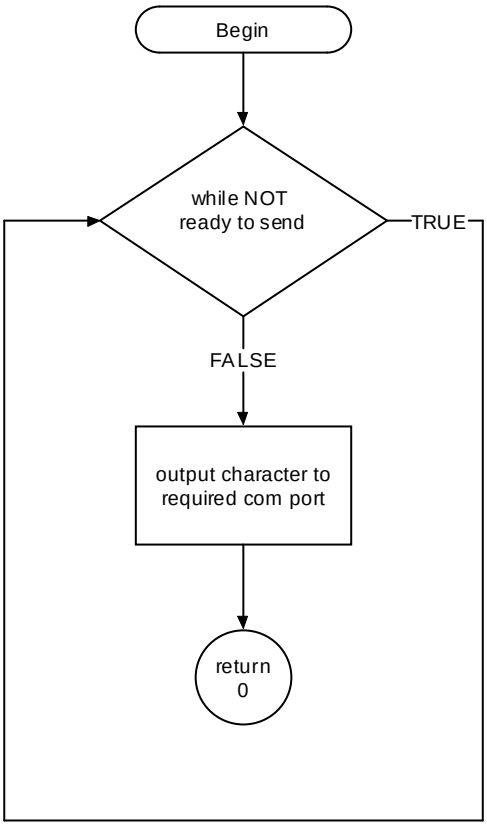
Third Handshake - NOT DLE & NOT SOH



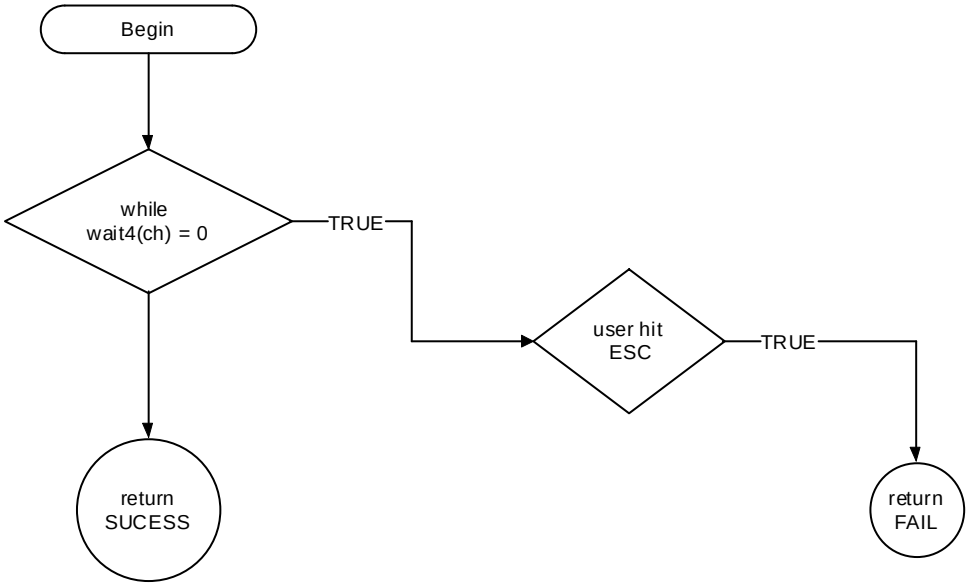
send

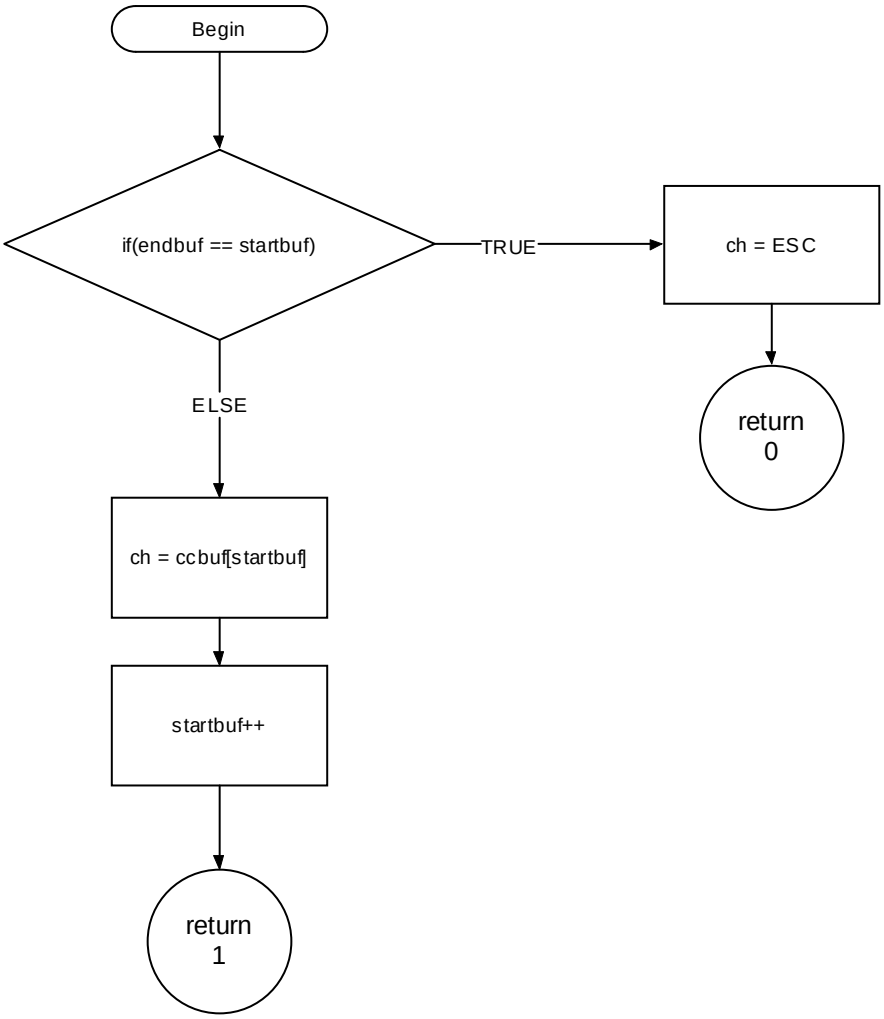


send_no_delay



wait_for_response



wait4

Functions for reading DEX audit data