

PRZEMYSŁOWY INSTYTUT AUTOMATYKI I POMIARÓW
MERA-PIAP
Al. Jerozolimskie 202 02-222 Warszawa Telefon 23-70-81

ZESPÓŁ UKŁADÓW I SYSTEMÓW STEROWANIA

440

BE 10

Główny wykonawca dr inż. W. Stańczak

Wykonawcy dr inż. W. Stańczak, mgr inż. M. Partyka
oraz T. Misztal i Jarosław Zajac, studenci Wydziału
Mechaniczno-Technologicznego PW /praca przejściowa/

Konsultant

Nr zlecenia S 1368

Implementacja oprogramowania MAP 3.0
i MMS
Etap 1. Rozpoznanie oprogramowania firmy
MODICON w nawiązaniu do zbioru usług
przewidzianych w MMS i Specyfikacji MAP 3.0
Identyfikacja narzędzi języka C użytych
do realizacji podstawowych usług

Zleceniodawca

Pracę rozpoczęto dnia 01.02.93

Kierownik Zespołu

dr inż. A. Syrczyński

zakończono dnia 15.06.93

Z-ca Dyrektora d/s
Badawczo-Rozwojowych

dr inż. J. Jabłkowski

Praca zawiera:

stron 50

rysunków

fotografii

tabel

tablic

załączników

Rozdzielnik - ilość egz: 3

Egz. 1 BOINTE

Egz. 2 ZSS

Egz. 3 ZSS

Egz. 4

Egz. 5

Egz. 6

Nr rejestr. 6967

Analiza deskryptorowa URZADZENIA AUTOMATYKI REGULACJI STEROWANIA:
SIEC LOKALNA + MAP + MMS 5 OPROGRAMOWANIE

Analiza dokumentacyjna

Sprawozdanie jest rezultatem rozpoznania oprogramowania MAP 3.0 i MMS firmy MODICON, zawiera wykaz i omówienie funkcji przewidzianych do realizacji usług MMS w pakiecie oprogramowania MMS - EASE.

Tytuły poprzednich sprawozdań

1. Badania i promocja sieci MAP. Etap 4.
Sprzężenie komputera standardu IBM z siecią MAP stałego stanowiska. Nr rejestr. 6910.

Narzędziami języka C wykorzystywanymi przez programistę tworzącego program (aplikacyjny) są funkcje biblioteczne oraz odpowiednio zorganizowane dane. Ich definicje umieszczane są zazwyczaj w tzw. plikach nagłówkowych (header lub include file).

Pakiet oprogramowania MMS - EASE firmy AEG - MODICON - COMPUTROL zawiera następujące biblioteki funkcji: ASN1.LIB, ASN1D.LIB, MMS.LIB, MMSACSE.LIB, MMSACSED.LIB, MMSD.LIB, SUICACSD.LIB, SUICACSE.LIB, U_ACSE.LIB, U_ACSED.LIB, U_CORE.LIB oraz U_CORED.LIB. Były one tworzone przy pomocy oprogramowania narzędziowego zawartego w pakiecie oprogramowania kompilatora języka C firmy Microsoft wersja 6.00A i są z nim w pełni kompatybilne. Oznacza to łatwość wykorzystania wspomnianego kompilatora i zawartych w nim standardowych funkcji języka C do tworzenia oprogramowania aplikacyjnego realizującego usługi MMS.

Praca zawiera wykaz i zwięzłe omówienie 220 podstawowych funkcji oraz dwu dodatkowych grup podstawowych funkcji, których użyto w pakiecie oprogramowania MMS - EASE firmy AEG - MODICON - COMPUTROL do realizacji usług przewidzianych w specyfikacji MMS. Informacje tu przytoczone uzyskano na podstawie przeprowadzonych eksperymentów, opierając się przy tym na dostarczonej wraz z MMS - EASE dokumentacją (która nota bene jest bardzo niekompletna i zawiera wiele błędów). Definicji poszczególnych typów danych (w szczególności struktur) w tym opracowaniu nie przytoczono, gdyż można je odszukać w 48 plikach nagłówkowych (łącznie 438554 bajty) - dostarczanych w ramach pakietu MMS - EASE (pliki: ASN1DEFS.H, CMPTRL_A.H, CSTK_OFF.H, CSTK_ON.H, FKEYDEFS.H, GEN_LIST.H, GLBERROR.H, GLBTYPES.H, GVALDEFS.H, LLCUSER.H, MEM_CHK.H, MMS_DEFS.H, MMS_DFUN.H, MMS_ERR.H, MMS_LLP.H, MMS_MP.H, MMS_MS.H, MMS_MV.H, MMS_PCON.H, MMS_PDOM.H, MMS_PERR.H, MMS_PEVN.H, MMS_PFIL.H, MMS_PJOU.H, MMS_POCS.H, MMS_PPRG.H, MMS_PSEM.H, MMS_PVAR.H, MMS_PVMD.H, MMS_UFUN.H, MMS_USER.H, MMS_USR.H, MMS_VCON.H, MMS_VDOM.H, MMS_VFIL.H, MMS_VPRG.H, MMS_VVAR.H, MMS_VVMD.H, MMSDEFS.H, MMSEFUN.H, MMSINTRN.H, MMSLLP.H, MMSOP_EN.H, SCRNDEFS.H, SUICACSE.H, SUIL.H, SYSINCS.H oraz USERDEFS.H).

Funkcje biblioteczne pakietu oprogramowania MMS - EASE dzielą się na dwie klasy. Pierwsza z nich służy do automatycznego wykonywania poszczególnych usług w wyniku zastosowania koncepcji tzw. VMD (Virtual Manufacturing Device). Druga klasa funkcji spełnia te same zadania, jednak realizacja poszczególnych usług jest rozbita na kolejne kroki. Dzięki temu użytkownik (programista) ma wgląd w przebieg realizacji programu.

W obydwu klasach stosowana jest ta sama konwencja - przekazywanie danych odbywa się przy użyciu PDU (Protocol Data Unit), które w języku C realizowane są jako struktury.

001. SHORT mp_abort(SHORT chan, SHORT reason)
 - kasuje przypisanie do kanału;
 SHORT chan - nr kanału;
 SHORT reason - ignorowane w przypadku ACSE.
002. MMSREQ_PEND * mp_ackevnot(SHORT chan,
 struct ackevnot_req_info *info_ptr)
 - przesyła PDU zlecenia AcknowledgeEventNotification;
 SHORT chan - numer kanału, którym zachodzi przesłanie;
 struct ackevnot_req_info *info_ptr - wskazanie na
 strukturę zawierającą dane PDU, które mają być
 wysłane.
003. MMSREQ_PEND * mp_altecm(SHORT chan,
 struct altecm_req_info *info_ptr)
 - przesyła PDU zlecenia AlterEventConditionMonitoring;
 SHORT chan - numer kanału, którym zachodzi przesłanie;
 struct acktecm_req_info *info_ptr - wskazanie na
 strukturę zawierającą dane PDU, które mają być
 wysłane.
004. MMSREQ_PEND * mp_altee(SHORT chan,
 struct altee_req_info *info_ptr)
 - przesyła PDU zlecenia AlterEventEnrollment;
 SHORT chan - numer kanału, którym zachodzi przesłanie;
 struct altee_req_info *info_ptr - wskazanie na
 strukturę zawierającą dane PDU, które mają być
 wysłane.
005. MMSREQ_PEND * mp_cancel(struct MMSREQ_PEND *req_ptr)
 - przesyła PDU zlecenia Cancel, dla odwołania
 uprzednio wysłanego zlecenia;
 struct MMSREQ_PEND *req_ptr - wskazanie na strukturę,
 którą zwrócono w wyniku wykonania funkcji
 typu mp_???? lub mv_???? zawierającej
 polecenie przesłania zlecenia.
006. MMSREQ_PEND * mp_conclude(SHORT chan)
 - przesyła PDU zlecenia Conclude w celu pozytywnego
 zakończenia przypisania do kanału;
 SHORT chan - numer kanału, którego przypisanie
 zakończono.

007. MMSREQ_PEND * mp_crepi(SHORT chan,
 struct crepi_req_info *info_ptr)
- przesyła PDU zlecenia CreateProgramInvocation;
SHORT chan - numer kanału, którym zachodzi przesłanie;
struct crepi_req_info *info_ptr - wskazanie na
strukturę zawierającą dane PDU, które mają
być wysłane.
008. MMSREQ_PEND * mp_defea(SHORT chan,
 struct defsem_req_info *info_ptr)
- przesyła PDU zlecenia DefineEventAction;
SHORT chan - numer kanału, którym zachodzi przesłanie;
struct defsem_req_info *info_ptr - wskazanie na
strukturę zawierającą dane PDU, które mają
być wysłane.
009. MMSREQ_PEND * mp_defec(SHORT chan,
 struct defsem_req_info *info_ptr)
- przesyła PDU zlecenia DefineEventCondition;
SHORT chan - numer kanału, którym zachodzi przesłanie;
struct defsem_req_info *info_ptr - wskazanie na
strukturę zawierającą dane PDU, które mają
być wysłane.
010. MMSREQ_PEND * mp_defee(SHORT chan,
 struct defsem_req_info *info_ptr)
- przesyła PDU zlecenia DefineEventEnrollment;
SHORT chan - numer kanału, którym zachodzi przesłanie;
struct defsem_req_info *info_ptr - wskazanie na
strukturę zawierającą dane PDU, które mają
być wysłane.
011. MMSREQ_PEND * mp_defsem(SHORT chan,
 struct defsem_req_info *info_ptr)
- przesyła PDU zlecenia DefineSemaphore;
SHORT chan - numer kanału, którym zachodzi przesłanie;
struct defsem_req_info *info_ptr - wskazanie na
strukturę zawierającą dane PDU, które mają
być wysłane.

012. MMSREQ_PEND * mp_defetype(SHORT chan,
 struct deftype_req_info *info_ptr)
- przesyła PDU zlecenia DefineNamedType;
SHORT chan - numer kanału, którym zachodzi przesłanie;
struct deftype_req_info *info_ptr - wskazanie na
strukturę zawierającą dane PDU, które mają
być wysłane.
013. MSREQ_PEND * mp_defvar(SHORT chan,
 struct defvar_req_info *info_ptr)
- przesyła PDU zlecenia DefineNamedVariable;
SHORT chan - numer kanału, którym zachodzi przesłanie;
struct defvar_req_info *info_ptr - wskazanie na
strukturę zawierającą dane PDU, które mają
być wysłane.
014. MMSREQ_PEND * mp_defvlist(SHORT chan,
 struct defvlist_req_info *info_ptr)
- przesyła PDU zlecenia DefineNamedVariableList;
SHORT chan - numer kanału, którym zachodzi przesłanie;
struct defvlist_req_info *info_ptr - wskazanie na
strukturę zawierającą dane PDU, które mają
być wysłane.
015. MMSREQ_PEND * mp_deldom(SHORT chan,
 struct deldom_req_info *info_ptr)
- przesyła PDU zlecenia DeleteDomain;
SHORT chan - numer kanału, którym zachodzi przesłanie;
struct deldom_req_info *info_ptr - wskazanie na
strukturę zawierającą dane PDU, które mają
być wysłane.
016. MMSREQ_PEND * mp_delea(SHORT chan,
 struct delea_req_info *info_ptr)
- przesyła PDU zlecenia DeleteEventAction;
SHORT chan - numer kanału, którym zachodzi przesłanie;
struct delea_req_info *info_ptr - wskazanie na
strukturę zawierającą dane PDU, które mają
być wysłane.

017. MMSREQ_PEND * mp_delec(SHORT chan,
 struct delec_req_info *info_ptr)
- przesyła PDU zlecenia DeleteEventCondition;
SHORT chan - numer kanału, którym zachodzi przesłanie;
struct delec_req_info *info_ptr - wskazanie na
strukturę zawierającą dane PDU, które mają
być wysłane.
018. MMSREQ_PEND * mp_delee(SHORT chan,
 struct delee_req_info *info_ptr)
- przesyła PDU zlecenia DeleteEventEnrollment;
SHORT chan - numer kanału, którym zachodzi przesłanie;
struct delee_req_info *info_ptr - wskazanie na
strukturę zawierającą dane PDU, które mają
być wysłane.
019. MMSREQ_PEND * mp_delpi(SHORT chan,
 struct delpi_req_info *info_ptr)
- przesyła PDU zlecenia DeleteProgramInvocation;
SHORT chan - numer kanału, którym zachodzi przesłanie;
struct delpi_req_info *info_ptr - wskazanie na
strukturę zawierającą dane PDU, które mają
być wysłane.
020. MMSREQ_PEND * mp_delsem(SHORT chan,
 struct delsem_req_info *info_ptr)
- przesyła PDU zlecenia DeleteSemaphore;
SHORT chan - numer kanału, którym zachodzi przesłanie;
struct delsem_req_info *info_ptr - wskazanie na
strukturę zawierającą dane PDU, które mają
być wysłane.
021. MMSREQ_PEND * mp_deltype(SHORT chan,
 struct deltype_req_info *info_ptr)
- przesyła PDU zlecenia DeleteNamedType;
SHORT chan - numer kanału, którym zachodzi przesłanie;
struct deltype_req_info *info_ptr - wskazanie na
strukturę zawierającą dane PDU, które mają
być wysłane.

022. MMSREQ_PEND * mp_delvar(SHORT chan,
 struct delvar_req_info *info_ptr)
- przesyła PDU zlecenia DeleteVariableAccess;
SHORT chan - numer kanału, którym zachodzi przesłanie;
struct delvar_req_info *info_ptr - wskazanie na
 strukturę zawierającą dane PDU, które mają
 być wysłane.
023. MMSREQ_PEND * mp_delvlist(SHORT chan,
 struct delvlist_req_info *info_ptr)
- przesyła PDU zlecenia DeleteNamedVariableList;
SHORT chan - numer kanału, którym zachodzi przesłanie;
struct delvlist_req_info *info_ptr - wskazanie na
 strukturę zawierającą dane PDU, które mają
 być wysłane.
024. MMSREQ_PEND * mp_download(SHORT chan,
 struct download_req_info *info_ptr)
- przesyła PDU zlecenia DownloadSegment;
SHORT chan - numer kanału, którym zachodzi przesłanie;
struct download_req_info *info_ptr - wskazanie na
 strukturę zawierającą dane PDU, które mają
 być wysłane.
025. MMSREQ_PEND * mp_evnot(SHORT chan,
 struct evnot_req_info *info_ptr)
- przesyła PDU zlecenia EventNotification;
SHORT chan - numer kanału, którym zachodzi przesłanie;
struct evnot_req_info *info_ptr - wskazanie na
 strukturę zawierającą dane PDU, które mają
 być wysłane.
026. MMSREQ_PEND * mp_fclose(SHORT chan,
 struct fclose_req_info *info_ptr)
- przesyła PDU zlecenia FileClose;
SHORT chan - numer kanału, którym zachodzi przesłanie;
struct fclose_req_info *info_ptr - wskazanie na
 strukturę zawierającą dane PDU, które mają
 być wysłane.

027. MMSREQ_PEND * mp_fdelete(SHORT chan,
 struct fdelete_req_info *info_ptr)
- przesyła PDU zlecenia FileDelete;
SHORT chan - numer kanału, którym zachodzi przesłanie;
struct fdelete_req_info *info_ptr - wskazanie na
 strukturę, która zawiera informacje o ścieżce
 dostępu do kasowanego pliku i jego nazwie.
028. MMSREQ_PEND * mp_fdir(SHORT chan,
 struct fdir_req_info *info_ptr)
- przesyła PDU zlecenia FileDirectory;
SHORT chan - numer kanału, którym zachodzi przesłanie;
struct fdir_req_info *info_ptr - wskazanie na
 strukturę zawierającą dane PDU, które mają
 być wysłane.
030. MMSREQ_PEND * mp_fopen(SHORT chan,
 struct fopen_req_info *info_ptr)
- przesyła PDU zlecenia FileOpen;
SHORT chan - numer kanału, którym zachodzi przesłanie;
struct fopen_req_info *info_ptr - wskazanie na
 strukturę zawierającą dane PDU, które mają
 być wysłane.
031. MMSREQ_PEND * mp_fread(SHORT chan,
 struct fread_req_info *info_ptr)
- przesyła PDU zlecenia FileRead;
SHORT chan - numer kanału, którym zachodzi przesłanie;
struct fread_req_info *info_ptr - wskazanie na
 strukturę zawierającą dane PDU, które mają
 być wysłane.
032. MMSREQ_PEND * mp_frename(SHORT chan,
 struct rename_req_info *info_ptr)
- przesyła PDU zlecenia FileRename;
SHORT chan - numer kanału, którym zachodzi przesłanie;
struct rename_req_info *info_ptr - wskazanie na
 strukturę zawierającą dane PDU, które mają
 być wysłane.

033. MMSREQ_PEND * mp_getaes(SHORT chan,
 struct getaes_req_info *info_ptr)
- przesyła PDU zlecenia GetAlarmEnrollmentSummary;
SHORT chan - numer kanału, którym zachodzi przesłanie;
struct getaes_req_info *info_ptr - wskazanie na
 strukturę zawierającą dane PDU, które mają
 być wysłane.
035. MMSREQ_PEND * mp_getas(SHORT chan,
 struct getas_req_info *info_ptr)
- przesyła PDU zlecenia GetAlarmSummary;
SHORT chan - numer kanału, którym zachodzi przesłanie;
struct getas_req_info *info_ptr - wskazanie na
 strukturę zawierającą dane PDU, które mają
 być wysłane.
036. MMSREQ_PEND * mp_getcl(SHORT chan,
 struct getcl_req_info *info_ptr)
- przesyła PDU zlecenia GetCapabilityList;
SHORT chan - numer kanału, którym zachodzi przesłanie;
struct getcl_req_info *info_ptr - wskazanie na
 strukturę zawierającą dane PDU, które mają
 być wysłane.
037. MMSREQ_PEND * mp_getdom(SHORT chan,
 struct getdom_req_info *info_ptr)
- przesyła PDU zlecenia GetDomainAttributes;
SHORT chan - numer kanału, którym zachodzi przesłanie;
struct getdom_req_info *info_ptr - wskazanie na
 strukturę zawierającą dane PDU, które mają
 być wysłane.
038. MMSREQ_PEND * mp_geteaa(SHORT chan,
 struct geteaa_req_info *info_ptr)
- przesyła PDU zlecenia GetEventActionAttributes;
SHORT chan - numer kanału, którym zachodzi przesłanie;
struct geteaa_req_info *info_ptr - wskazanie na
 strukturę zawierającą dane PDU, które mają
 być wysłane.

039. MMSREQ_PEND * mp_geteca(SHORT chan,
 struct geteca_req_info *info_ptr)
- przesyła PDU zlecenia GetEventConditionAttributes;
SHORT chan - numer kanału, którym zachodzi przesłanie;
struct geteca_req_info *info_ptr - wskazanie na
strukturę zawierającą dane PDU, które mają
być wysłane.
040. MMSREQ_PEND * mp_geteea(SHORT chan,
 struct geteea_req_info *info_ptr)
- przesyła PDU zlecenia GetEventEnrollmentAttributes;
SHORT chan - numer kanału, którym zachodzi przesłanie;
struct geteea_req_info *info_ptr - wskazanie na
strukturę zawierającą dane PDU, które mają
być wysłane.
041. MMSREQ_PEND * mp_getpi(SHORT chan,
 struct getpi_req_info *info_ptr)
- przesyła PDU zlecenia GetProgramInvocationAttributes;
SHORT chan - numer kanału, którym zachodzi przesłanie;
struct getpi_req_info *info_ptr - wskazanie na
strukturę zawierającą dane PDU, które mają
być wysłane.
042. MMSREQ_PEND * mp_gettype(SHORT chan,
 struct gettype_req_info *info_ptr)
- przesyła PDU zlecenia GetNamedTypeAttributes;
SHORT chan - numer kanału, którym zachodzi przesłanie;
struct gettype_req_info *info_ptr - wskazanie na
strukturę zawierającą dane PDU, które mają
być wysłane.
043. MMSREQ_PEND * mp_getvar(SHORT chan,
 struct getvar_req_info *info_ptr)
- przesyła PDU zlecenia GetVariableAccessAttributes;
SHORT chan - numer kanału, którym zachodzi przesłanie;
struct getvar_req_info *info_ptr - wskazanie na
strukturę zawierającą dane PDU, które mają
być wysłane.

044. MMSREQ_PEND * mp_getvlist(SHORT chan,
 struct getvlist_req_info *info_ptr)
- przesyła PDU zlecenia GetNamedVariableListAttributes;
SHORT chan - numer kanału, którym zachodzi przesłanie;
struct getvlist_req_info *info_ptr - wskazanie na
 strukturę zawierającą dane PDU, które mają
 być wysłane.
045. MMSREQ_PEND * mp_ident(SHORT chan)
- przesyła PDU zlecenia Identify;
SHORT chan - numer kanału, którym zachodzi przesłanie;
046. SHORT mp_info(SHORT chan,
 struct info_req_info *info_ptr)
- przesyła PDU zlecenia InformationReport;
SHORT chan - numer kanału, którym zachodzi przesłanie;
struct info_req_info *info_ptr - wskazanie na
 strukturę zawierającą dane PDU, które mają
 być wysłane.
047. MMSREQ_PEND * mp_initdown(SHORT chan,
 struct initdown_req_info *info_ptr)
- przesyła PDU zlecenia InitiateDownloadSequence;
SHORT chan - numer kanału, którym zachodzi przesłanie;
struct initdown_req_info *info_ptr - wskazanie na
 strukturę zawierającą dane PDU, które mają
 być wysłane.
048. MMSREQ_PEND * mp_initupl(SHORT chan,
 struct initupl_req_info *info_ptr)
- przesyła PDU zlecenia InitiateUploadSequence;
SHORT chan - numer kanału, którym zachodzi przesłanie;
struct initupl_req_info *info_ptr - wskazanie na
 strukturę zawierającą dane PDU, które mają
 być wysłane.

049. MMSREQ_PEND * mp_input(SHORT chan,
 struct input_req_info *info_ptr)
- przesyła PDU zlecenia Input;
SHORT chan - numer kanału, którym zachodzi przesłanie;
struct input_req_info *info_ptr - wskazanie na
 strukturę zawierającą dane PDU, które mają
 być wysłane.
050. MMSREQ_PEND * mp_jcreate(SHORT chan,
 struct jcreate_req_info *info_ptr)
- przesyła PDU zlecenia CreateJournal;
SHORT chan - numer kanału, którym zachodzi przesłanie;
struct jcreate_req_info *info_ptr - wskazanie na
 strukturę zawierającą dane PDU, które mają
 być wysłane.
051. MMSREQ_PEND * mp_jdelete(SHORT chan,
 struct jdelete_req_info *info_ptr)
- przesyła PDU zlecenia DeleteJournal;
SHORT chan - numer kanału, którym zachodzi przesłanie;
struct jdelete_req_info *info_ptr - wskazanie na
 strukturę zawierającą dane PDU, które mają
 być wysłane.
052. MMSREQ_PEND * mp_jinit(SHORT chan,
 struct jinit_req_info *info_ptr)
- przesyła PDU zlecenia InitializeJournal;
SHORT chan - numer kanału, którym zachodzi przesłanie;
struct jinit_req_info *info_ptr - wskazanie na
 strukturę zawierającą dane PDU, które mają
 być wysłane.
053. MMSREQ_PEND * mp_jread(SHORT chan,
 struct jread_req_info *info_ptr)
- przesyła PDU zlecenia ReadJournal;
SHORT chan - numer kanału, którym zachodzi przesłanie;
struct jread_req_info *info_ptr - wskazanie na
 strukturę zawierającą dane PDU, które mają
 być wysłane.

054. MMSREQ_PEND * mp_jstat(SHORT chan,
 struct jstat_req_info *info_ptr)
- przesyła PDU zlecenia ReportJournalStatus;
SHORT chan - numer kanału, którym zachodzi przesłanie;
struct jstat_req_info *info_ptr - wskazanie na
 strukturę zawierającą dane PDU, które mają
 być wysłane.
055. MMSREQ_PEND * mp_jwrite(SHORT chan,
 struct jwrite_req_info *info_ptr)
- przesyła PDU zlecenia WriteJournal;
SHORT chan - numer kanału, którym zachodzi przesłanie;
struct jwrite_req_info *info_ptr - wskazanie na
 strukturę zawierającą dane PDU, które mają
 być wysłane.
056. MMSREQ_PEND * mp_kill(SHORT chan,
 struct kill_req_info *info_ptr)
- przesyła PDU zlecenia Kill;
SHORT chan - numer kanału, którym zachodzi przesłanie;
struct kill_req_info *info_ptr - wskazanie na
 strukturę zawierającą dane PDU, które mają
 być wysłane.
057. MMSREQ_PEND * mp_loaddom(SHORT chan,
 struct loaddom_req_info *info_ptr)
- przesyła PDU zlecenia LoadDomainContent;
SHORT chan - numer kanału, którym zachodzi przesłanie;
struct loaddom_req_info *info_ptr - wskazanie na
 strukturę zawierającą dane PDU, które mają
 być wysłane.
058. MMSREQ_PEND * mp_namelist(SHORT chan,
 struct namelist_req_info *info_ptr)
- przesyła PDU zlecenia GetNameList;
SHORT chan - numer kanału, którym zachodzi przesłanie;
struct namelist_req_info *info_ptr - wskazanie na
 strukturę zawierającą dane PDU, które mają
 być wysłane.

059. MMSREQ_PEND * mp_obtfile(SHORT chan,
 struct obtfile_req_info *info_ptr)
- przesyła PDU zlecenia ObtainFile;
SHORT chan - numer kanału, którym zachodzi przesłanie;
struct obtfile_req_info *info_ptr - wskazanie na
 strukturę zawierającą dane PDU, które mają
 być wysłane.
060. MMSREQ_PEND * mp_output(SHORT chan,
 struct output_req_info *info_ptr)
- przesyła PDU zlecenia Output;
SHORT chan - numer kanału, którym zachodzi przesłanie;
struct output_req_info *info_ptr - wskazanie na
 strukturę zawierającą dane PDU, które mają
 być wysłane.
061. MMSREQ_PEND * mp_rddwl(SHORT chan,
 struct rddwl_req_info *info_ptr)
- przesyła PDU zlecenia RequestDomainDownload;
SHORT chan - numer kanału, którym zachodzi przesłanie;
struct output_req_info *info_ptr - wskazanie na
 strukturę zawierającą dane PDU, które mają
 być wysłane.
062. MMSREQ_PEND * mp_rdupl(SHORT chan,
 struct rdupl_req_info *info_ptr)
- przesyła PDU zlecenia RequestDomainUpload;
SHORT chan - numer kanału, którym zachodzi przesłanie;
struct rdupl_req_info *info_ptr - wskazanie na
 strukturę zawierającą dane PDU, które mają
 być wysłane.
063. MMSREQ_PEND * mp_read(SHORT chan,
 struct read_req_info *info_ptr)
- przesyła PDU zlecenia Read;
SHORT chan - numer kanału, którym zachodzi przesłanie;
struct read_req_info *info_ptr - wskazanie na
 strukturę zawierającą dane PDU, które mają
 być wysłane.

064. MMSREQ_PEND * mp_relctrl(SHORT chan,
 struct relctrl_req_info *info_ptr)
- przesyła PDU zlecenia RelinquishControl;
SHORT chan - numer kanału, którym zachodzi przesłanie;
struct relctrl_req_info *info_ptr - wskazanie na
 strukturę zawierającą dane PDU, które mają
 być wysłane.
065. MMSREQ_PEND * mp_rename(SHORT chan,
 struct rename_req_info *info_ptr)
- przesyła PDU zlecenia Rename;
SHORT chan - numer kanału, którym zachodzi przesłanie;
struct rename_req_info *info_ptr - wskazanie na
 strukturę zawierającą dane PDU, które mają
 być wysłane.
066. MMSREQ_PEND * mp_repeas(SHORT chan,
 struct repeas_req_info *info_ptr)
- przesyła PDU zlecenia ReportEventActionStatus;
SHORT chan - numer kanału, którym zachodzi przesłanie;
struct repeas_req_info *info_ptr - wskazanie na
 strukturę zawierającą dane PDU, które mają
 być wysłane.
067. MMSREQ_PEND * mp_repcs(SHORT chan,
 struct repecs_req_info *info_ptr)
- przesyła PDU zlecenia ReportEventConditionStatus;
SHORT chan - numer kanału, którym zachodzi przesłanie;
struct repecs_req_info *info_ptr - wskazanie na
 strukturę zawierającą dane PDU, które mają
 być wysłane.
068. MMSREQ_PEND * mp_repees(SHORT chan,
 struct repees_req_info *info_ptr)
- przesyła PDU zlecenia ReportEventEnrollmentStatus;
SHORT chan - numer kanału, którym zachodzi przesłanie;
struct repees_req_info *info_ptr - wskazanie na
 strukturę zawierającą dane PDU, które mają
 być wysłane.

069. MMSREQ_PEND * mp_reset(SHORT chan,
 struct reset_req_info *info_ptr)
- przesyła PDU zlecenia Reset;
SHORT chan - numer kanału, którym zachodzi przesłanie;
struct reset_req_info *info_ptr - wskazanie na
strukturę zawierającą dane PDU, które mają
być wysłane.
070. MMSREQ_PEND * mp_resume(SHORT chan,
 struct resume_req_info *info_ptr)
- przesyła PDU zlecenia Resume;
SHORT chan - numer kanału, którym zachodzi przesłanie;
struct resume_req_info *info_ptr - wskazanie na
strukturę zawierającą dane PDU, które mają
być wysłane.
071. MMSREQ_PEND * mp_rsentry(SHORT chan,
 struct rsentry_req_info *info_ptr)
- przesyła PDU zlecenia ReportSemaphoreEntryStatus;
SHORT chan - numer kanału, którym zachodzi przesłanie;
struct rsentry_req_info *info_ptr - wskazanie na
strukturę zawierającą dane PDU, które mają
być wysłane.
072. MMSREQ_PEND * mp_rspool(SHORT chan,
 struct rspool_req_info *info_ptr)
- przesyła PDU zlecenia ReportPoolSemaphoreStatus;
SHORT chan - numer kanału, którym zachodzi przesłanie;
struct rspool_req_info *info_ptr - wskazanie na
strukturę zawierającą dane PDU, które mają
być wysłane.
073. MMSREQ_PEND * mp_rsstat(SHORT chan,
 struct rsstat_req_info *info_ptr)
- przesyła PDU zlecenia ReportSemaphoreStatus;
SHORT chan - numer kanału, którym zachodzi przesłanie;
struct rsstat_req_info *info_ptr - wskazanie na
strukturę zawierającą dane PDU, które mają
być wysłane.

074. MMSREQ_PEND * mp_start(SHORT chan,
 struct start_req_info *info_ptr)
- przesyła PDU zlecenia Start;
SHORT chan - numer kanału, którym zachodzi przesłanie;
struct start_req_info *info_ptr - wskazanie na
 strukturę zawierającą dane PDU, które mają
 być wysłane.
075. MMSREQ_PEND * mp_status(SHORT chan,
 struct status_req_info *info_ptr)
- przesyła PDU zlecenia Status;
SHORT chan - numer kanału, którym zachodzi przesłanie;
struct status_req_info *info_ptr - wskazanie na
 strukturę zawierającą dane PDU, które mają
 być wysłane.
076. MMSREQ_PEND * mp_stop(SHORT chan, struct
 stop_req_info *info_ptr)
- przesyła PDU zlecenia Stop;
SHORT chan - numer kanału, którym zachodzi przesłanie;
struct stop_req_info *info_ptr - wskazanie na
 strukturę zawierającą dane PDU, które mają
 być wysłane.
077. MMSREQ_PEND * mp_storedom(SHORT chan,
 struct storedom_req_info *info_ptr)
- przesyła PDU zlecenia StoreDomainContent;
SHORT chan - numer kanału, którym zachodzi przesłanie;
struct storedom_req_info *info_ptr - wskazanie na
 strukturę zawierającą dane PDU, które mają
 być wysłane.
078. MMSREQ_PEND * mp_takectrl(SHORT chan,
 struct takectrl_req_info *info_ptr)
- przesyła PDU zlecenia TakeControl;
SHORT chan - numer kanału, którym zachodzi przesłanie;
struct takectrl_req_info *info_ptr - wskazanie na
 strukturę zawierającą dane PDU, które mają
 być wysłane.

079. MMSREQ_PEND * mp_termdown(SHORT chan,
 struct termdown_req_info *info_ptr)
- przesyła PDU zlecenia TerminateDownloadSequence;
SHORT chan - numer kanału, którym zachodzi przesłanie;
struct termdown_req_info *info_ptr - wskazanie na
strukturę zawierającą dane PDU, które mają
być wysłane.
080. MMSREQ_PEND * mp_termupl(SHORT chan,
 struct termupl_req_info *info_ptr)
- przesyła PDU zlecenia TerminateUploadSequence;
SHORT chan - numer kanału, którym zachodzi przesłanie;
struct termupl_req_info *info_ptr - wskazanie na
strukturę zawierającą dane PDU, które mają
być wysłane.
081. MMSREQ_PEND * mp_trige(SHORT chan,
 struct ackevnot_req_info *info_ptr)
- przesyła PDU zlecenia TriggerEvent;
SHORT chan - numer kanału, którym zachodzi przesłanie;
struct trige_req_info *info_ptr - wskazanie na
strukturę zawierającą dane PDU, które mają
być wysłane.
082. MMSREQ_PEND * mp_upload(SHORT chan,
 struct upload_req_info *info_ptr)
- przesyła PDU zlecenia UploadSegmentRequest;
SHORT chan - numer kanału, którym zachodzi przesłanie;
struct upload_req_info *info_ptr - wskazanie na
strukturę zawierającą dane PDU, które mają
być wysłane.
083. MMSREQ_PEND * mp_ustatus(SHORT chan,
 struct ustatus_req_info *info_ptr)
- przesyła PDU zlecenia UnsolicitedStatus;
SHORT chan - numer kanału, którym zachodzi przesłanie;
struct ustatus_req_info *info_ptr - wskazanie na
strukturę zawierającą dane PDU, które mają
być wysłane.

084. MMSREQ_PEND * mp_write(SHORT chan,
 struct write_req_info *info_ptr)
- przesyła PDU zlecenia Write;
SHORT chan - numer kanału, którym zachodzi przesłanie;
struct write_req_info *info_ptr - wskazanie na
strukturę zawierającą dane PDU, które mają
być wysłane.
085. SHORT mp_ackevnot_resp(struct MMSREQ_IND *req_ptr)
- przesyła PDU z (pozytywnym) potwierdzeniem wykonania
zlecenia AcknowledgeEventNotification;
struct MMSREQ_IND *req_ptr - wskazanie na strukturę
zwróconą przez funkcję u_ackevnot_ind.
086. SHORT mp_altecm_resp(struct MMSREQ_IND *req_ptr)
- przesyła PDU z (pozytywnym) potwierdzeniem wykonania
zlecenia AlterEventConditionMonitoring;
struct MMSREQ_IND *req_ptr - wskazanie na strukturę
zwróconą przez funkcję u_altecm_ind.
087. SHORT mp_altee_resp(struct MMSREQ_IND *req_ptr,
 struct altee_resp_info *info_ptr)
- przesyła PDU z (pozytywnym) potwierdzeniem wykonania
zlecenia AlterEventEnrollment;
struct MMSREQ_IND *req_ptr - wskazanie na strukturę
zwróconą przez funkcję u_altee_ind;
struct altee_resp_info *info_ptr - wskazanie na
strukturę zawierającą dane o zleceniu.
088. SHORT mp_cancel_resp(struct MMSREQ_IND *req_ptr)
- przesyła PDU z (pozytywnym) potwierdzeniem wykonania
zlecenia Cancel;
struct MMSREQ_IND *req_ptr - wskazanie na strukturę
zwróconą przez funkcję u_????_ind.
089. SHORT mp_conclude_resp(struct MMSREQ_IND *req_ptr)
- przesyła PDU z (pozytywnym) potwierdzeniem wykonania
zlecenia Conclude;
struct MMSREQ_IND *req_ptr - wskazanie na strukturę
zwróconą przez funkcję u_conclude_ind.

090. SHORT mp_crepi_resp(struct MMSREQ_IND *req_ptr)
 - przesyła PDU z (pozytywnym) potwierdzeniem wykonania
 zlecenia CreateProgramInvocation;
 struct MMSREQ_IND *req_ptr - wskazanie na strukturę
 zwróconą przez funkcję u_crepi_ind.
091. SHORT mp_defea_resp(struct MMSREQ_IND *req_ptr)
 - przesyła PDU z (pozytywnym) potwierdzeniem wykonania
 zlecenia DefineEventAction;
 struct MMSREQ_IND *req_ptr - wskazanie na strukturę
 zwróconą przez funkcję u_defea_ind.
092. SHORT mp_defec_resp(struct MMSREQ_IND *req_ptr)
 - przesyła PDU z (pozytywnym) potwierdzeniem wykonania
 zlecenia DefineEventCondition;
 struct MMSREQ_IND *req_ptr - wskazanie na strukturę
 zwróconą przez funkcję u_defec_ind.
093. SHORT mp_defee_resp(struct MMSREQ_IND *req_ptr)
 - przesyła PDU z (pozytywnym) potwierdzeniem wykonania
 zlecenia DefineEventEnrollment;
 struct MMSREQ_IND *req_ptr - wskazanie na strukturę
 zwróconą przez funkcję u_defee_ind.
094. SHORT mp_defsem_resp(struct MMSREQ_IND *req_ptr)
 - przesyła PDU z (pozytywnym) potwierdzeniem wykonania
 zlecenia DefineSemaphore;
 struct MMSREQ_IND *req_ptr - wskazanie na strukturę
 zwróconą przez funkcję u_defsem_ind.
095. SHORT mp_deftype_resp(struct MMSREQ_IND *req_ptr)
 - przesyła PDU z (pozytywnym) potwierdzeniem wykonania
 zlecenia DefineNamedType;
 struct MMSREQ_IND *req_ptr - wskazanie na strukturę
 zwróconą przez funkcję u_deftype_ind.
096. SHORT mp_defvar_resp(struct MMSREQ_IND *req_ptr)
 - przesyła PDU z (pozytywnym) potwierdzeniem wykonania
 zlecenia DefineNamedVariable;
 struct MMSREQ_IND *req_ptr - wskazanie na strukturę
 zwróconą przez funkcję u_defvar_ind.

097. SHORT mp_defvlist_resp(struct MMSREQ_IND *req_ptr)
 - przesyła PDU z (pozytywnym) potwierdzeniem wykonania zlecenia DefineNamedVariableList;
 struct MMSREQ_IND *req_ptr - wskazanie na strukturę zwróconą przez funkcję u_defvlist_ind.
098. SHORT mp_deldom_resp(struct MMSREQ_IND *req_ptr)
 - przesyła PDU z (pozytywnym) potwierdzeniem wykonania zlecenia DeleteDomain;
 struct MMSREQ_IND *req_ptr - wskazanie na strukturę zwróconą przez funkcję u_deldom_ind.
099. SHORT mp_delea_resp(struct MMSREQ_IND *req_ptr,
 struct delea_resp_info *info_ptr)
 - przesyła PDU z (pozytywnym) potwierdzeniem wykonania zlecenia DeleteEventAction;
 struct MMSREQ_IND *req_ptr - wskazanie na strukturę zwracaną przez funkcję u_delea_ind;
 struct delea_resp_info *info_ptr - wskazanie na strukturę zawierającą dane o zleceniu.
100. SHORT mp_delec_resp(struct MMSREQ_IND *req_ptr,
 struct delec_resp_info *info_ptr)
 - przesyła PDU z (pozytywnym) potwierdzeniem wykonania zlecenia DeleteEventCondition;
 struct MMSREQ_IND *req_ptr - wskazanie na strukturę zwracaną przez funkcję u_delec_ind;
 struct delec_resp_info *info_ptr - wskazanie na strukturę zawierającą dane o zleceniu.
101. SHORT mp_delee_resp(struct MMSREQ_IND *req_ptr,
 struct delee_resp_info *info_ptr)
 - przesyła PDU z (pozytywnym) potwierdzeniem wykonania zlecenia DeleteEventEnrollment;
 struct MMSREQ_IND *req_ptr - wskazanie na strukturę zwracaną przez funkcję u_delea_ind;
 struct delee_resp_info *info_ptr - wskazanie na strukturę zawierającą dane o zleceniu.

102. SHORT mp_delpi_resp(struct MMSREQ_IND *req_ptr)
- przesyła PDU z (pozytywnym) potwierdzeniem wykonania
zlecenia DeleteProgramInvocation;
struct MMSREQ_IND *req_ptr - wskazanie na strukturę
zwróconą przez funkcję u_delpi_ind.
103. SHORT mp_delsem_resp(struct MMSREQ_IND *req_ptr)
- przesyła PDU z (pozytywnym) potwierdzeniem wykonania
zlecenia DeleteSemaphore;
struct MMSREQ_IND *req_ptr - wskazanie na strukturę
zwracaną przez funkcję u_delsem_ind.
104. SHORT mp_deltype_resp(struct MMSREQ_IND *req_ptr,
struct deltype_resp_info *info_ptr)
- przesyła PDU z (pozytywnym) potwierdzeniem wykonania
zlecenia DeleteNamedType;
struct MMSREQ_IND *req_ptr - wskazanie na strukturę
zwracaną przez funkcję u_deltype_ind;
struct deltype_resp_info *info_ptr - wskazanie na
strukturę zawierającą informacje o zleceniu.
105. SHORT mp_delvar_resp(struct MMSREQ_IND *req_ptr,
struct delvar_resp_info *info_ptr)
- przesyła PDU z (pozytywnym) potwierdzeniem wykonania
zlecenia DeleteVariableAccess;
struct MMSREQ_IND *req_ptr - wskazanie na strukturę
zwracaną przez funkcję u_delvar_ind.
struct delvar_resp_info *info_ptr - wskazanie na
strukturę zawierającą informacje o zleceniu.
106. SHORT mp_delvlist_resp(struct MMSREQ_IND *req_ptr,
struct delvlist_resp_info *info_ptr)
- przesyła PDU z (pozytywnym) potwierdzeniem wykonania
zlecenia DeleteNamedVariableList;
struct MMSREQ_IND *req_ptr - wskazanie na strukturę
zwracaną przez funkcję u_delvlist_ind;
struct delvlist_resp_info *info_ptr - wskazanie na
strukturę zawierającą informacje o zleceniu.

107. SHORT mp_download_resp(struct MMSREQ_IND *req_ptr,
 struct download_resp_info *info_ptr)
- przesyła PDU z (pozytywnym) potwierdzeniem wykonania
 zlecenia DownloadSegment;
struct MMSREQ_IND *req_ptr - wskazanie na strukturę
 zwracaną przez funkcję u_download_ind.
struct download_resp_info *info_ptr - wskazanie na
 strukturę zawierającą informacje o zleceniu.
108. SHORT mp_fclose_resp(struct MMSREQ_IND *req_ptr)
- przesyła PDU z (pozytywnym) potwierdzeniem wykonania
 zlecenia FileClose;
struct MMSREQ_IND *req_ptr - wskazanie na strukturę
 zwracaną przez funkcję u_fclose_ind.
109. SHORT mp_fdelete_resp(struct MMSREQ_IND *req_ptr)
- przesyła PDU z (pozytywnym) potwierdzeniem wykonania
 zlecenia FileDelete;
struct MMSREQ_IND *req_ptr - wskazanie na strukturę
 zwróconą przez funkcję u_fdelete_ind.
110. SHORT mp_fdir_resp(struct MMSREQ_IND *req_ptr,
 struct fdir_resp_info *info_ptr)
- przesyła PDU z (pozytywnym) potwierdzeniem wykonania
 zlecenia FileDirectory;
struct MMSREQ_IND *req_ptr - wskazanie na strukturę
 zwracaną przez funkcję u_fclose_ind;
struct fdir_resp_info *info_ptr - wskazanie na
 strukturę zawierającą informacje o zleceniu.
111. SHORT mp_fopen_resp(struct MMSREQ_IND *req_ptr,
 struct fopen_resp_info *info_ptr)
- przesyła PDU z (pozytywnym) potwierdzeniem wykonania
 zlecenia FileOpen;
struct MMSREQ_IND *req_ptr - wskazanie na strukturę
 zwracaną przez funkcję u_fopen_ind;
struct fopen_resp_info *info_ptr - wskazanie na
 strukturę zawierającą dane o zleceniu.

112. SHORT mp_fread_resp(struct MMSREQ_IND *req_ptr,
 struct fread_resp_info *info_ptr)
- przesyła PDU z (pozytywnym) potwierdzeniem wykonania
 zlecenia FileRead;
struct MMSREQ_IND *req_ptr - wskazanie na strukturę
 zwracaną przez funkcję u_fread_ind;
struct fread_resp_info *info_ptr - wskazanie na
 strukturę zawierającą dane o zleceniu.
113. SHORT mp_frename_resp(struct MMSREQ_IND *req_ptr)
- przesyła PDU z (pozytywnym) potwierdzeniem wykonania
 zlecenia FileRename;
struct MMSREQ_IND *req_ptr - wskazanie na strukturę
 zwracaną przez funkcję u_frename_ind.
114. SHORT mp_getaes_resp(struct MMSREQ_IND *req_ptr,
 struct gataes_resp_info *info_ptr)
- przesyła PDU z (pozytywnym) potwierdzeniem wykonania
 zlecenia GetAlarmEnrollmentSummary;
struct MMSREQ_IND *req_ptr - wskazanie na strukturę
 zwracaną przez funkcję u_getaes_ind.
struct gataes_resp_info *info_ptr - wskazanie na
 strukturę zawierającą informacje o zleceniu.
115. SHORT mp_getas_resp(struct MMSREQ_IND *req_ptr,
 struct getas_resp_info *info_ptr)
- przesyła PDU z (pozytywnym) potwierdzeniem wykonania
 zlecenia GetAlarmSummary;
struct MMSREQ_IND *req_ptr - wskazanie na strukturę
 zwracaną przez funkcję u_getas_ind;
struct getas_resp_info *info_ptr - wskazanie na
 strukturę zawierającą informacje o zleceniu.
116. SHORT mp_getcl_resp(struct MMSREQ_IND *req_ptr,
 struct getcl_resp_info *info_ptr)
- przesyła PDU z (pozytywnym) potwierdzeniem wykonania
 zlecenia GetCapabilityList;
struct MMSREQ_IND *req_ptr - wskazanie na strukturę
 zwracaną przez funkcję u_getcl_ind;
struct getcl_resp_info *info_ptr - wskazanie na
 strukturę zawierającą informacje o zleceniu.

117. SHORT mp_getdom_resp(struct MMSREQ_IND *req_ptr,
 struct getdom_resp_info *info_ptr)
- przesyła PDU z (pozytywnym) potwierdzeniem wykonania
 zlecenia GetDomainAttributes;
struct MMSREQ_IND *req_ptr - wskazanie na strukturę
 zwracaną przez funkcję u_getdom_ind;
struct getdom_resp_info *info_ptr - wskazanie na
 strukturę zawierającą informacje o zleceniu.
118. SHORT mp_geteaa_resp(struct MMSREQ_IND *req_ptr,
 struct geteaa_resp_info *info_ptr)
- przesyła PDU z (pozytywnym) potwierdzeniem wykonania
 zlecenia GetEventActionAttributes;
struct MMSREQ_IND *req_ptr - wskazanie na strukturę
 zwracaną przez funkcję u_geteaa_ind;
struct geteaa_resp_info *info_ptr - wskazanie na
 strukturę zawierającą informacje o zleceniu.
119. SHORT mp_geteca_resp(struct MMSREQ_IND *req_ptr,
 struct geteca_resp_info *info_ptr)
- przesyła PDU z (pozytywnym) potwierdzeniem wykonania
 zlecenia GetEventConditionAttributes;
struct MMSREQ_IND *req_ptr - wskazanie na strukturę
 zwróconą przez funkcję u_geteca_ind;
struct geteca_resp_info *info_ptr - wskazanie na
 strukturę zawierającą informacje o zleceniu.
120. mp_geteea_resp(struct MMSREQ_IND *req_ptr,
 struct geteea_resp_info *info_ptr)
- przesyła PDU z (pozytywnym) potwierdzeniem wykonania
 zlecenia GetEventEnrollmentAttributes;
struct MMSREQ_IND *req_ptr - wskazanie na strukturę
 zwróconą przez funkcję u_geteea_ind;
struct geteea_resp_info *info_ptr - wskazanie na
 strukturę zawierającą informacje o zleceniu.
121. SHORT mp_getpi_resp(struct MMSREQ_IND *req_ptr,
 struct getpi_resp_info *info_ptr)
- przesyła PDU z (pozytywnym) potwierdzeniem wykonania
 zlecenia GetProgramInvocationAttributes;
struct MMSREQ_IND *req_ptr - wskazanie na strukturę
 zwracaną przez funkcję u_getpi_ind;
struct getpi_resp_info *info_ptr - wskazanie na
 strukturę zawierającą informacje o zleceniu.

122. SHORT mp_gettype_resp(struct MMSREQ_IND *req_ptr,
 struct gettype_resp_info *info_ptr)
- przesyła PDU z (pozytywnym) potwierdzeniem wykonania
 zlecenia GetNamedTypeAttributes;
struct MMSREQ_IND *req_ptr - wskazanie na strukturę
 zwracaną przez funkcję u_gettype_ind;
struct gettype_resp_info *info_ptr - wskazanie na
 strukturę zawierającą informacje o zleceniu.
123. SHORT mp_getvar_resp(struct MMSREQ_IND *req_ptr,
 struct getvar_resp_info *info_ptr)
- przesyła PDU z (pozytywnym) potwierdzeniem wykonania
 zlecenia GetVariableAccessAttributes;
struct MMSREQ_IND *req_ptr - wskazanie na strukturę
 zwracaną przez funkcję u_getvar_ind;
struct getvar_resp_info *info_ptr - wskazanie na
 strukturę zawierającą informacje o zleceniu.
124. SHORT mp_getvlist_resp(struct MMSREQ_IND *req_ptr,
 struct getvar_resp_info *info_ptr)
- przesyła PDU z (pozytywnym) potwierdzeniem wykonania
 zlecenia GetNamedVariableListAttributes;
struct MMSREQ_IND *req_ptr - wskazanie na strukturę
 zwracaną przez funkcję u_getvlist_ind;
struct getvar_resp_info *info_ptr - wskazanie na
 strukturę zawierającą informacje o zleceniu.
125. SHORT mp_ident_resp(struct MMSREQ_IND *req_ptr,
 struct ident_resp_info *info_ptr)
- przesyła PDU z (pozytywnym) potwierdzeniem wykonania
 zlecenia Identify;
struct MMSREQ_IND *req_ptr - wskazanie na strukturę
 zwracaną przez funkcję u_ident_ind;
struct ident_resp_info *info_ptr - wskazanie na
 strukturę zawierającą informacje o zleceniu.
126. SHORT mp_initdown_resp(struct MMSREQ_IND *req_ptr)
- przesyła PDU z (pozytywnym) potwierdzeniem wykonania
 zlecenia InitiateDownloadSequence;
struct MMSREQ_IND *req_ptr - wskazanie na strukturę
 zwracaną przez funkcję u_initdown_ind.

127. SHORT mp_initupl_resp(struct MMSREQ_IND *req_ptr,
 struct initupl_resp_info *info_ptr)
 - przesyła PDU z (pozytywnym) potwierdzeniem wykonania
 zlecenia InitiateUploadSequence;
 struct MMSREQ_IND *req_ptr - wskazanie na strukturę
 zwracaną przez funkcję u_initupl_ind;
 struct initupl_resp_info *info_ptr - wskazanie na
 strukturę zawierającą informacje o zleceniu.
128. SHORT mp_input_resp(struct MMSREQ_IND *req_ptr,
 struct input_resp_info *info_ptr)
 - przesyła PDU z (pozytywnym) potwierdzeniem wykonania
 zlecenia Input;
 struct MMSREQ_IND *req_ptr - wskazanie na strukturę
 zwracaną przez funkcję u_input_ind;
 struct input_resp_info *info_ptr - wskazanie na
 strukturę zawierającą informacje o zleceniu.
129. SHORT mp_jcreate_resp(struct MMSREQ_IND *req_ptr)
 - przesyła PDU z (pozytywnym) potwierdzeniem wykonania
 zlecenia CreateJournal;
 struct MMSREQ_IND *req_ptr - wskazanie na strukturę
 zwracaną przez funkcję u_jcreate_ind.
130. SHORT mp_jdelete_resp(struct MMSREQ_IND *req_ptr)
 - przesyła PDU z (pozytywnym) potwierdzeniem wykonania
 zlecenia DeleteJournal;
 struct MMSREQ_IND *req_ptr - wskazanie na strukturę
 zwracaną przez funkcję u_jdelete_ind.
131. SHORT mp_jinit_resp(struct MMSREQ_IND *req_ptr,
 struct jinit_resp_info *info_ptr)
 - przesyła PDU z (pozytywnym) potwierdzeniem wykonania
 zlecenia InitializeJournal;
 struct MMSREQ_IND *req_ptr - wskazanie na strukturę
 zwracaną przez funkcję u_jinit_ind;
 struct jinit_resp_info *info_ptr - wskazanie na
 strukturę zawierającą informacje o zleceniu.
132. SHORT mp_jread_resp(struct MMSREQ_IND *req_ptr,
 struct jread_resp_info *info_ptr)
 - przesyła PDU z (pozytywnym) potwierdzeniem wykonania
 zlecenia ReadJournal;

- struct MMSREQ_IND *req_ptr - wskazanie na strukturę zwracaną przez funkcję u_jread_ind;
 struct jread_resp_info *info_ptr - wskazanie na strukturę zawierającą informacje o zleceniu.
133. SHORT mp_jstat_resp(struct MMSREQ_IND *req_ptr,
 struct jstat_resp_info *info_ptr)
 - przesyła PDU z (pozytywnym) potwierdzeniem wykonania zlecenia ReportJournalStatus;
 struct MMSREQ_IND *req_ptr - wskazanie na strukturę zwracaną przez funkcję u_jstat_ind;
 struct jstat_resp_info *info_ptr - wskazanie na strukturę zawierającą informacje o zleceniu.
134. SHORT mp_jwrite_resp(struct MMSREQ_IND *req_ptr)
 - przesyła PDU z (pozytywnym) potwierdzeniem wykonania zlecenia WriteJournal;
 struct MMSREQ_IND *req_ptr - wskazanie na strukturę zwracaną przez funkcję u_jwrite_ind.
135. SHORT mp_kill_resp(struct MMSREQ_IND *req_ptr)
 - przesyła PDU z (pozytywnym) potwierdzeniem wykonania zlecenia Kill;
 struct MMSREQ_IND *req_ptr - wskazanie na strukturę zwróconą przez funkcję u_kill_ind.
136. SHORT mp_loaddom_resp(struct MMSREQ_IND *req_ptr)
 - przesyła PDU z (pozytywnym) potwierdzeniem wykonania zlecenia LoadDomainContent;
 struct MMSREQ_IND *req_ptr - wskazanie na strukturę zwróconą przez funkcję u_loaddom_ind.
137. SHORT mp_namelist_resp(struct MMSREQ_IND *req_ptr,
 struct namelist_resp_info *info_ptr)
 - przesyła PDU z (pozytywnym) potwierdzeniem wykonania zlecenia GetNameList;
 struct MMSREQ_IND *req_ptr - wskazanie na strukturę zwracaną przez funkcję u_namelist_ind;
 struct namelist_resp_info *info_ptr - wskazanie na strukturę zawierającą dane o zleceniu.
138. SHORT mp_obtfile_resp(struct MMSREQ_IND *req_ptr)
 - przesyła PDU z (pozytywnym) potwierdzeniem wykonania zlecenia ObtainFile;
 struct MMSREQ_IND *req_ptr - wskazanie na strukturę

zwracaną przez funkcję `u_obtfile_ind`.

139. `SHORT mp_output_resp(struct MMSREQ_IND *req_ptr)`
- przesyła PDU z (pozytywnym) potwierdzeniem wykonania zlecenia Output;
`struct MMSREQ_IND *req_ptr` - wskazanie na strukturę zwracaną przez funkcję `u_fclose_ind`.
140. `SHORT mp_rddwn_resp(struct MMSREQ_IND *req_ptr)`
- przesyła PDU z (pozytywnym) potwierdzeniem wykonania zlecenia RequestDomainDownload;
`struct MMSREQ_IND *req_ptr` - wskazanie na strukturę zwróconą przez funkcję `u_rddwn_ind`.
141. `SHORT mp_rdupl_resp(struct MMSREQ_IND *req_ptr)`
- przesyła PDU z (pozytywnym) potwierdzeniem wykonania zlecenia RequestDomainUpload;
`struct MMSREQ_IND *req_ptr` - wskazanie na strukturę zwróconą przez funkcję `u_rdupl_ind`.
142. `SHORT mp_read_resp(struct MMSREQ_IND *req_ptr,`
`struct read_resp_info *info_ptr)`
- przesyła PDU z (pozytywnym) potwierdzeniem wykonania zlecenia Read;
`struct MMSREQ_IND *req_ptr` - wskazanie na strukturę zwracaną przez funkcję `u_read_ind`;
`struct read_resp_info *info_ptr` - wskazanie na strukturę zawierającą informacje o zleceniu.
143. `SHORT mp_relctrl_resp(struct MMSREQ_IND *req_ptr)`
- przesyła PDU z (pozytywnym) potwierdzeniem wykonania zlecenia RelinquishControl;
`struct MMSREQ_IND *req_ptr` - wskazanie na strukturę zwracaną przez funkcję `u_relctr_ind`.
144. `SHORT mp_rename_resp(struct MMSREQ_IND *req_ptr)`
- przesyła PDU z (pozytywnym) potwierdzeniem wykonania zlecenia Rename;
`struct MMSREQ_IND *req_ptr` - wskazanie na strukturę zwracaną przez funkcję `u_rename_ind`.
145. `SHORT mp_repeas_resp(struct MMSREQ_IND *req_ptr,`
`struct repeas_resp_info *info_ptr)`
- przesyła PDU z (pozytywnym) potwierdzeniem wykonania zlecenia ReportEventActionStatus;

- struct MMSREQ_IND *req_ptr - wskazanie na strukturę zwracaną przez funkcję u_repeas_ind;
 struct repeas_resp_info *info_ptr - wskazanie na strukturę zawierającą informacje o zleceniu.
146. SHORT mp_repecs_resp(struct MMSREQ_IND *req_ptr,
 struct repecs_resp_info *info_ptr)
 - przesyła PDU z (pozytywnym) potwierdzeniem wykonania zlecenia ReportEventConditionStatus;
 struct MMSREQ_IND *req_ptr - wskazanie na strukturę zwracaną przez funkcję u_repecs_ind;
 struct repecs_resp_info *info_ptr - wskazanie na strukturę zawierającą informacje o zleceniu.
147. SHORT mp_repees_resp(struct MMSREQ_IND *req_ptr,
 struct repees_resp_info *info_ptr)
 - przesyła PDU z (pozytywnym) potwierdzeniem wykonania zlecenia ReportEventEnrollmentStatus;
 struct MMSREQ_IND *req_ptr - wskazanie na strukturę zwracaną przez funkcję u_repees_ind;
 struct repees_resp_info *info_ptr - wskazanie na strukturę zawierającą informacje o zleceniu.
148. SHORT mp_reset_resp(struct MMSREQ_IND *req_ptr)
 - przesyła PDU z (pozytywnym) potwierdzeniem wykonania zlecenia Reset;
 struct MMSREQ_IND *req_ptr - wskazanie na strukturę zwróconą przez funkcję u_reset_ind.
149. SHORT mp_resume_resp(struct MMSREQ_IND *req_ptr)
 - przesyła PDU z (pozytywnym) potwierdzeniem wykonania zlecenia Resume;
 struct MMSREQ_IND *req_ptr - wskazanie na strukturę zwróconą przez funkcję u_resume_ind.
150. SHORT mp_rsendy_resp(struct MMSREQ_IND *req_ptr,
 struct rsendy_resp_info *info_ptr)
 - przesyła PDU z (pozytywnym) potwierdzeniem wykonania zlecenia ReportSemaphoreEntryStatus;
 struct MMSREQ_IND *req_ptr - wskazanie na strukturę zwracaną przez funkcję u_rsendy_ind;
 struct rsendy_resp_info *info_ptr - wskazanie na strukturę zawierającą informacje o zleceniu.

151. SHORT mp_rspool_resp(struct MMSREQ_IND *req_ptr,
 struct rspool_resp_info *info_ptr)
- przesyła PDU z (pozytywnym) potwierdzeniem wykonania
zlecenia ReportPoolSemaphoreStatus;
struct MMSREQ_IND *req_ptr - wskazanie na strukturę
zwracaną przez funkcję u_rspool_ind;
struct rspool_resp_info *info_ptr - wskazanie na
strukturę zawierającą informacje o zleceniu.
152. SHORT mp_rsstat_resp(struct MMSREQ_IND *req_ptr,
 struct rsstat_resp_info *info_ptr)
- przesyła PDU z (pozytywnym) potwierdzeniem wykonania
zlecenia ReportSemaphoreStatus;
struct MMSREQ_IND *req_ptr - wskazanie na strukturę
zwracaną przez funkcję u_rsstat_ind;
struct rsstat_resp_info *info_ptr - wskazanie na
strukturę zawierającą informacje o zleceniu.
153. SHORT mp_start_resp(struct MMSREQ_IND *req_ptr)
- przesyła PDU z (pozytywnym) potwierdzeniem wykonania
zlecenia Start;
struct MMSREQ_IND *req_ptr - wskazanie na strukturę
zwróconą przez funkcję u_start_ind.
154. SHORT mp_status_resp(struct MMSREQ_IND *req_ptr,
 struct status_resp_info *info_ptr)
- przesyła PDU z (pozytywnym) potwierdzeniem wykonania
zlecenia Status;
struct MMSREQ_IND *req_ptr - wskazanie na strukturę
zwróconą przez funkcję u_status_ind;
struct status_resp_info *info_ptr - wskazanie na
strukturę zawierającą informacje o zleceniu.
155. SHORT mp_stop_resp(struct MMSREQ_IND *req_ptr)
- przesyła PDU z (pozytywnym) potwierdzeniem wykonania
zlecenia Stop;
struct MMSREQ_IND *req_ptr - wskazanie na strukturę
zwróconą przez funkcję u_stop_ind.
156. SHORT mp_storedom_resp(struct MMSREQ_IND *req_ptr)
- przesyła PDU z (pozytywnym) potwierdzeniem wykonania
zlecenia StoreDomainContent;
struct MMSREQ_IND *req_ptr - wskazanie na strukturę
zwróconą przez funkcję u_storedom_ind.

157. SHORT mp_takectrl_resp(struct MMSREQ_IND *req_ptr,
 struct takectrl_resp_info *info_ptr)
- przesyła PDU z (pozytywnym) potwierdzeniem wykonania
 zlecenia TakeControl;
struct MMSREQ_IND *req_ptr - wskazanie na strukturę
 zwracaną przez funkcję u_takectrl_ind;
struct takectrl_resp_info *info_ptr - wskazanie na
 strukturę zawierającą informacje o zleceniu.
158. SHORT mp_termdown_resp(struct MMSREQ_IND *req_ptr)
- przesyła PDU z (pozytywnym) potwierdzeniem wykonania
 zlecenia TerminateDownloadSequence;
struct MMSREQ_IND *req_ptr - wskazanie na strukturę
 zwróconą przez funkcję u_termdown_ind.
159. SHORT mp_termupl_resp(struct MMSREQ_IND *req_ptr)
- przesyła PDU z (pozytywnym) potwierdzeniem wykonania
 zlecenia TerminateUploadSequence;
struct MMSREQ_IND *req_ptr - wskazanie na strukturę
 zwróconą przez funkcję u_termupl_ind.
160. SHORT mp_trige_resp(struct MMSREQ_IND *req_ptr)
- przesyła PDU z (pozytywnym) potwierdzeniem wykonania
 zlecenia TriggerEvent;
struct MMSREQ_IND *req_ptr - wskazanie na strukturę
 zwracaną przez funkcję u_trige_ind.
161. SHORT mp_upload_resp(struct MMSREQ_IND *req_ptr,
 struct upload_resp_info *info_ptr)
- przesyła PDU z (pozytywnym) potwierdzeniem wykonania
 zlecenia UploadSegmentRequest;
struct MMSREQ_IND *req_ptr - wskazanie na strukturę
 zwróconą przez funkcję u_upload_ind;
struct upload_resp_info *info_ptr - wskazanie na
 strukturę zawierającą informacje o zleceniu.
162. SHORT mp_write_resp(struct MMSREQ_IND *req_ptr,
 struct write_resp_info *info_ptr)
- przesyła PDU z (pozytywnym) potwierdzeniem wykonania
 zlecenia Write;
struct MMSREQ_IND *req_ptr - wskazanie na strukturę
 zwróconą przez funkcję u_write_ind;
struct write_resp_info *info_ptr - wskazanie na
 strukturę zawierającą informacje o zleceniu.

163. VOID u_mp_????_conf(MMSREQ_PEND *req_ptr)
 - jest to grupa funkcji (w jej skład wchodzi również funkcja u_mp_conclude_done(SHORT,USHORT)) służąca do potwierdzenia zleceń ujętych funkcjami typu mp_?????. Wszystkie z nich (poza omówionymi dalej, oddzielnie u_mp_conclude_done oraz u_mp_cancel_conf) mają przedstawiony wyżej format prototypu;
 MMSREQ_PEND *req_ptr - wskazanie na strukturę zwróconą w wyniku wykonania odpowiedniej, uprzednio wywołanej funkcji typu mp_????.

Poniżej zestawiono poszczególne funkcje typu u_mp_????_conf i odpowiadające im zlecenia - usługi MMS:

u_mp_ackevnot_conf(MMSREQ_PEND *) - AcknowledgeEventNotification
 u_mp_altecm_conf(MMSREQ_PEND *) - AlterEventConditionMonitoring;
 u_mp_altee_conf(MMSREQ_PEND *) - AlterEventEnrollment;
 u_mp_cancel_conf(MMSREQ_PEND *,UBYTE,struct err_info *) - Cancel
 u_mp_conclude_conf(MMSREQ_PEND *) - Conclude;
 u_mp_conclude_done(SHORT,USHORT) - Conclude;
 u_mp_crepi_conf(MMSREQ_PEND *) - CreateProgramInvocation;
 u_mp_defea_conf(MMSREQ_PEND *) - DefineEventAction;
 u_mp_defec_conf(MMSREQ_PEND *) - DefineEventCondition;
 u_mp_defee_conf(MMSREQ_PEND *) - DefineEventEnrollment;
 u_mp_defsem_conf(MMSREQ_PEND *) - DefineSemaphore;
 u_mp_deftype_conf(MMSREQ_PEND *) - DefineNamedType;
 u_mp_defvar_conf(MMSREQ_PEND *) - DefineNamedVariable;
 u_mp_defvlist_conf(MMSREQ_PEND *) - DefineNamedVariableList;
 u_mp_deldom_conf(MMSREQ_PEND *) - DeleteDomain;
 u_mp_delea_conf(MMSREQ_PEND *) - DeleteEventAction;
 u_mp_delec_conf(MMSREQ_PEND *) - DeleteEventCondition;
 u_mp_delee_conf(MMSREQ_PEND *) - DeleteEventEnrollment;
 u_mp_delpi_conf(MMSREQ_PEND *) - DeleteProgramInvocation;
 u_mp_delsem_conf(MMSREQ_PEND *) - DeleteSemaphore;
 u_mp_deltype_conf(MMSREQ_PEND *) - DeleteNamedType;
 u_mp_delvar_conf(MMSREQ_PEND *) - DeleteVariableAccess;
 u_mp_delvlist_conf(MMSREQ_PEND *) - DeleteNamedVariableList;
 u_mp_download_conf(MMSREQ_PEND *) - DownloadSegment;
 u_mp_evnot_conf(MMSREQ_PEND *) - EventNotification;
 u_mp_fclose_conf(MMSREQ_PEND *) - FileClose;
 u_mp_fdelete_conf(MMSREQ_PEND *) - FileDelete;
 u_mp_fdir_conf(MMSREQ_PEND *) - FileDirectory;

u_mp_fopen_conf(MMSREQ_PEND *) - FileOpen;
 u_mp_fread_conf(MMSREQ_PEND *) - FileRead;
 u_mp_frename_conf(MMSREQ_PEND *) - FileRename;
 u_mp_getaes_conf(MMSREQ_PEND *) - GetAlarmEnrollmentSummary;
 u_mp_getas_conf(MMSREQ_PEND *) - GetAlarmSummary;
 u_mp_getcl_conf(MMSREQ_PEND *) - GetCapabilityList;
 u_mp_getdom_conf(MMSREQ_PEND *) - GetDomainAttributes;
 u_mp_geteaa_conf(MMSREQ_PEND *) - GetEventActionAttributes;
 u_mp_geteca_conf(MMSREQ_PEND *) - GetEventConditionAttributes;
 u_mp_geteea_conf(MMSREQ_PEND *) - GetEventEnrollmentAttributes;
 u_mp_getpi_conf(MMSREQ_PEND *) - GetProgramInvocationAttributes;
 u_mp_gettype_conf(MMSREQ_PEND *) - GetNamedTypeAttributes;
 u_mp_getvar_conf(MMSREQ_PEND *) - GetVariableAccessAttributes;
 u_mp_getvlist_conf(MMSREQ_PEND *) - GetNamedVariableListAttribut

u_mp_ident_conf(MMSREQ_PEND *) - Identify;
 u_mp_initdown_conf(MMSREQ_PEND *) - InitiateDownloadSequence;
 u_mp_initupl_conf(MMSREQ_PEND *) - InitiateUploadSequence;
 u_mp_input_conf(MMSREQ_PEND *) - Input;
 u_mp_jcreate_conf(MMSREQ_PEND *) - CreateJournal;
 u_mp_jdelete_conf(MMSREQ_PEND *) - DeleteJournal;
 u_mp_jinit_conf(MMSREQ_PEND *) - InitializeJournal;
 u_mp_jread_conf(MMSREQ_PEND *) - ReadJournal;
 u_mp_jstat_conf(MMSREQ_PEND *) - ReportJournalStatus;
 u_mp_jwrite_conf(MMSREQ_PEND *) - WriteJournal;
 u_mp_kill_conf(MMSREQ_PEND *) - Kill;
 u_mp_loaddom_conf(MMSREQ_PEND *) - LoadDomainContent;
 u_mp_namelist_conf(MMSREQ_PEND *) - GetNameList;
 u_mp_obtfile_conf(MMSREQ_PEND *) - ObtainFile;
 u_mp_output_conf(MMSREQ_PEND *) - Output;
 u_mp_rddwn_conf(MMSREQ_PEND *) - RequestDomainDownload;
 u_mp_rdupl_conf(MMSREQ_PEND *) - RequestDomainUpload;
 u_mp_read_conf(MMSREQ_PEND *) - Read;
 u_mp_relctrl_conf(MMSREQ_PEND *) - RelinquishControl;
 u_mp_rename_conf(MMSREQ_PEND *) - Rename;
 u_mp_repeas_conf(MMSREQ_PEND *) - ReportEventActionStatus;
 u_mp_repecs_conf(MMSREQ_PEND *) - ReportEventConditionStatus;
 u_mp_repees_conf(MMSREQ_PEND *) - ReportEventEnrollmentStatus;
 u_mp_reset_conf(MMSREQ_PEND *) - Reset;
 u_mp_resume_conf(MMSREQ_PEND *) - Resume;
 u_mp_rsenry_conf(MMSREQ_PEND *) - ReportSemaphoreEntryStatus;
 u_mp_rspool_conf(MMSREQ_PEND *) - ReportPoolSemaphoreStatus;
 u_mp_rsstat_conf(MMSREQ_PEND *) - ReportSemaphoreStatus;

u_mp_start_conf(MMSREQ_PEND *) - Start;
 u_mp_status_conf(MMSREQ_PEND *) - Status;
 u_mp_stop_conf(MMSREQ_PEND *) - Stop;
 u_mp_storedom_conf(MMSREQ_PEND *) - StoreDomainContent;
 u_mp_takectrl_conf(MMSREQ_PEND *) - TakeControl;
 u_mp_termdown_conf(MMSREQ_PEND *) - TerminateDownloadSequence;
 u_mp_termupl_conf(MMSREQ_PEND *) - TerminateUploadSequence;
 u_mp_trige_conf(MMSREQ_PEND *) - TriggerEvent;
 u_mp_upload_conf(MMSREQ_PEND *) - UploadSegmentRequest;
 u_mp_write_conf(MMSREQ_PEND *) - Write;

164. VOID u_mp_cancel_conf(MMSREQ_PEND *req_ptr,
 UBYTE errdata_pres, struct err_info *err_ptr)
 - służy do przetwarzania informacji zwracanej przez
 zlecenie Cancel;
 MMSREQ_PEND *req_ptr - wskazanie na strukturę
 zwróconą w wyniku wykonania uprzednio
 wywołanej funkcji mp_cancel;
 UBYTE errdata_pres - numer błędu powstały przy
 realizacji zlecenia Cancel (w przypadku braku
 błędu = 0);
 struct err_info *err_ptr -wskazanie na strukturę
 zawierającą opis błędu związany z realizacją
 zlecenia Cancel (a nie uprzedniej usługi, do
 której Cancel ma zastosowanie).

165. VOID u_mp_conclude_done(SHORT chan, USHORT ret_code)
 - służy do rozstrzygnięcia sytuacji powstałej po
 zrealizowaniu zlecenia Conclude (po wykonaniu
 funkcji mp_conclude);
 SHORT chan - numer kanału, który właśnie został
 zwolniony;
 USHORT ret_code - kod zwrócony z warstwy LLP. Jeżeli
 zero, znaczy że zwolnienie kanału powiodło
 się. Jeżeli różny od zera, to zwolnienie
 kanału nie powiodło się i połączenie należy
 przerwać stosując funkcję mp_abort.

166. SHORT mp_cancel_err(MMSREQ_IND *req_ptr, SHORT class,
SHORT code)
- przesyła PDU o błędzie w przypadku trudności ze
zaprzestaniem wykonywania uprzednio
sformułowanego zlecenia (tzn. wykonania
zlecenia Cancel).
MMSREQ_IND *req_ptr - wskazanie na strukturę
odpowiadającą zleceniu, którego wykonanie być
zaniechane (jest to to samo wskazanie, które
było użyte dla funkcji u_????_ind);
SHORT class - numer klasy błędów wg. ISO 9506;
SHORT code - kod błędu.
167. SHORT mp_conclude_err(MMSREQ_IND *req_ptr,
SHORT class, SHORT code)
- przesyła PDU o błędzie w przypadku, w którym nie
należy wykonywać zlecenia Cancel;
MMSREQ_IND *req_ptr - wskazanie na strukturę
odpowiadającą zleceniu, którego wykonanie być
zaniechane (jest to to samo wskazanie, które
było użyte dla funkcji u_conclude_ind);
SHORT class - numer klasy błędów wg. ISO 9506;
SHORT code - kod błędu.
168. SHORT mp_err_resp(MMSREQ_IND *req_ptr, SHORT class,
SHORT code)
- przesyła PDU o błędzie występującym w przypadku, w
którym wykonywano zlecenia różne od Cancel;
MMSREQ_IND *req_ptr - wskazanie na strukturę
odpowiadającą zleceniu, którego wykonanie być
zaniechane (jest to to samo wskazanie, które
było użyte dla funkcji u_????_ind);
SHORT class - numer klasy błędów wg. ISO 9506;
SHORT code - kod błędu.
169. SHORT mp_init_err(MMSREQ_IND *req_ptr, SHORT class,
SHORT code)
- przesyła PDU o błędzie w przypadku, w którym
zlecenie Initiate zostało odrzucone;
MMSREQ_IND *req_ptr - wskazanie na strukturę
odpowiadającą zleceniu, którego wykonanie być
zaniechane (jest to to samo wskazanie, które
było użyte dla funkcji u_init_ind);
SHORT class - numer klasy błędów wg. ISO 9506;
SHORT code - kod błędu.

170. SHORT mp_reject_conf(MMSREQ_IND *req_ptr,
 struct reject_resp_info *info_ptr)
- odrzuca przychodzące potwierdzenie;
MMSREQ_IND *req_ptr - wskazanie na strukturę
 odpowiadającą zleceniu, którego wykonanie być
 zaniechane (jest to to samo wskazanie, które
 było przekazane funkcji u_mp_????_conf);
struct reject_resp_info *info_ptr - wskazanie na
 strukturę, która zawiera informacje
 szczegółowe dla PDU zlecenia Reject.
171. SHORT mp_reject_ind(MMSREQ_IND *req_ptr,
 struct reject_resp_info *info_ptr)
- przesyła PDU zlecenia Reject zamiast pozytywnego lub
 negatywnego potwierdzenia;
MMSREQ_IND *req_ptr - wskazanie na strukturę
 odpowiadającą zleceniu, którego wykonanie być
 zaniechane (jest to to samo wskazanie, które
 było przekazane funkcji u_????_ind);
struct reject_resp_info *info_ptr - wskazanie na
 strukturę, która zawiera informacje
 szczegółowe dla PDU zlecenia Reject.
172. MMSREQ_PEND * mv_download(SHORT chan, BYTE *dom_name,
 SHORT src_type, SHORT num_of_capab,
 BYTE *capab_list[num_of_capab], LONG src_len,
 UBYTE *src)
- wykonuje całą sekwencję związaną z usługą Download
 (tzn. InitiateDownload, wielokrotnie
 DownloadSegment oraz TerminateDownload);
SHORT chan - kanał, którym wysyła się PDU;
BYTE *dom_name - wskazanie na nazwę obiektu
 podlegającego usłudze Download;
SHORT src_type - typ obiektu podlegającego usłudze
 Download;
SHORT num_of_capab - liczba pozycji w tablicy
 capab_list[num_of_capab] (por. niżej);
BYTE *capab_list[num_of_capab] - tablica wskazań do
 ciągów znaków (tekstów) kończonych zerem
 związanych z obiektami VMD (Virtual
 Manufacturing Device);
LONG src_len - długość (w bajtach) obiektu
 podlegającego usłudze Download;
UBYTE *src - wskazanie na obiekt podlegający usłudze
 Download.

173. MMSREQ_PEND * mv_fclose(SHORT chan, LONG frsmid)
 - wykonuje usługę FileClose odwołując się do VMD;
 SHORT chan - kanał, w którym plik był otwarty;
 LONG frsmid - stan VMD otrzymany podczas wykonywania FileOpen.
174. MMSREQ_PEND * mv_fcopy(SHORT chan, BYTE *source,
 SHORT source_len, BYTE *dest, SHORT dest_len)
 - kopiowanie pliku przez odwołanie się do VMD;
 SHORT chan - kanał, który posłuży do kopiowania pliku;
 BYTE *source - wskazanie na nazwę ścieżki i nazwę pliku kopiowanego;
 SHORT source_len - długość (w bajtach) pliku i nazwy ścieżki;
 BYTE *dest - wskazanie na nazwę ścieżki i nazwę pliku po skopiowaniu;
 SHORT dest_len - długość (w bajtach) pliku po skopiowaniu i jego ścieżki.
175. MMSREQ_PEND * mv_fopen(SHORT chan, BYTE *path,
 SHORT path_len, ULONG init_pos)
 - otwieranie pliku (FileOpen) przez odwołanie się do VMD;
 SHORT chan - kanał, który posłuży do wykonania usługi;
 BYTE *path - nazwa ścieżki i pliku;
 SHORT path_len - długość w bajtach nazwy ścieżki i pliku;
 ULONG init_pos - początkowa pozycja w pliku, od której później rozpocznie działanie usługa FileRead.
176. MMSREQ_PEND * mv_fread(SHORT chan, SHORT frsmid,
 BYTE *dest_ptr, LONG num_bytes)
 - wykonanie usługi FileRead przez odwołanie się do VMD;
 SHORT chan - kanał, który posłuży do wykonania usługi;
 SHORT frsmid - stan File Read State Machine otrzymany podczas otwierania pliku;
 BYTE *dest_ptr - wskazanie na bufor pamięci, gdzie segment pliku będzie przechowywany;
 LONG num_bytes - liczba bajtów czytanych. Musi być ona równa liczbie bajtów w buforze dest_ptr.

177. SHORT mv_info(SHORT chan, struct object_name *vname,
struct object_name *tname, BYTE *src_data)
- wykonanie usługi InformationReport przez odwołanie się do VMD;
SHORT chan - kanał, który posłuży do wykonania usługi;
struct object_name *vname - wskazanie na strukturę zawierającą nazwę zmiennej włączaną do PDU usługi InformationReport;
struct object_name *tname - wskazanie na strukturę zawierającą lokalną nazwę zmiennej. Jeżeli argument ten = NULL, to MMS-EASE używa informacji z bazy danych VMD;
BYTE *src_data - wskazanie na miejsce włączenia informacji w PDU dla InformationReport.
178. SHORT mv_infovars(SHORT chan, SHORT num_vars,
struct mv_var_desc *var_desc)
- wykonanie usługi InformationReport przez odwołanie się do VMD (por. funkcja mv_info);
SHORT chan - kanał, który posłuży do wykonania usługi;
SHORT num_vars - liczba zmiennych włączonych do PDU. Musi być ona równa liczbie struktur typu mv_var_desc, które są wskazywane przez var_desc;
struct mv_var_desc *var_desc - wskazanie na początek tablicy struktur typu mv_var_desc, zawierających specyfikacje zmiennych włączanych do PDU dla InformationReport.
179. SHORT mv_info_report(SHORT chan, SHORT num_of_vars,
struct var_acc_spec *reqinfo,
struct mv_info_req_info *mv_info)
- wykonanie usługi InformationReport przez odwołanie się do VMD (por. funkcje mv_info oraz mv_infovars);
SHORT chan - kanał, który posłuży do wykonania usługi;
SHORT num_of_vars - liczba zmiennych włączonych do PDU. Musi być ona równa liczbie struktur typu mv_var_desc, które są wskazywane przez var_desc;

- struct var_acc_spec *reqinfo - wskazanie na tablicę struktur zawierających informacje, które zmienne są włączane do PDU dla InformationReport;
- struct mv_info_req_info *mv_info - wskazanie na tablicę struktur zawierających szczegółowe informacje o metodzie kodowania poszczególnych zmiennych.
180. MMSREQ_PEND * mv_init(SHORT chan, BYTE *partner)
 - wykonanie usługi Initiate przez odwołanie się do VMD;
 SHORT chan - kanał, który posłuży do wykonania usługi;
 BYTE *partner - wskazanie na nazwę (AR Name) procesu.
181. MMSREQ_PEND * mv_read(SHORT chan,
 struct object_name *vname,
 struct object_name *tname)
 - wykonanie usługi Read przez odwołanie się do VMD;
 SHORT chan - kanał, który posłuży do wykonania usługi;
 struct object_name *vname - wskazanie na strukturę specyfikującą czytane zmienne;
 struct object_name *tname - wskazanie na strukturę specyfikującej zmienne lokalne odniesione do zmiennych opisanych w vname;
 BYTE *dest_data - wskazanie na miejsce, w którym dane należy zapisać po otrzymaniu potwierdzenia dla usługi Read.
182. MMSREQ_PEND * mv_readvars(SHORT chan, SHORT num_vars,
 struct mv_vardesc *var_desc)
 - wykonanie usługi Read przez odwołanie się do VMD (por. funkcja mv_read);
 SHORT chan - kanał, który posłuży do wykonania usługi;
 SHORT num_var - liczba zmiennych czytanych w usłudze Read. Musi być ona równa liczbie struktur typu mv_vardesc, które są wskazywane przez var_desc;
 struct mv_vardesc *var_desc - wskazanie na początek tablicy struktur zawierających informacje o czytanych zmiennych.

183. MMSREQ_PEND * mv_read_variables(SHORT chan,
SHORT num_var, struct read_req_info *priminfo,
struct mv_read_req_info *vminfo)
- wykonanie usługi Read przez odwołanie się do VMD
(por. funkcje mv_read oraz mv_readvars);
SHORT chan - kanał, który posłuży do wykonania
usługi;
SHORT num_var - liczba zmiennych czytanych w usłudze
Read;
struct read_req_info *priminfo - wskazanie na
strukturę specyfikującą sposób interpretacji
danych;
struct mv_read_req_info *vminfo - wskazanie na
początek tablicy struktury zawierających
informację o tym, które zmienne mają być
czytane.
184. MMSREQ_PEND * mv_upload(SHORT chan, BYTE *dom_name,
SHORT dest_type, LONG dest_len, UBYTE *dest)
- wykonanie usługi Upload przez odwołanie się do VMD;
SHORT chan - kanał, który posłuży do wykonania
usługi;
BYTE *dom_name - wskazanie na nazwę domeny
podlegającej usłudze Upload;
SHORT dest_type - typ domeny;
LONG dest_len - rozmiar domeny (w bajtach);
UBYTE *dest - wskazanie na miejsce wczytania.
185. MMSREQ_PEND * mv_write(SHORT chan,
struct object_name *vname,
struct object_name *tname, BYTE *src_data)
- wykonanie usługi Write przez odwołanie się do VMD;
SHORT chan - kanał, który posłuży do wykonania
usługi;
struct object_name *vname - wskazanie na strukturę
specyfikującą zapisywane zmienne;
struct object_name *tname - wskazanie na strukturę
specyfikującą zmienne lokalne odniesione do
zmiennych opisanych w vname;
BYTE *src_data - wskazanie na miejsce, w którym należy
szukać danych do zapisania.

186. MMSREQ_PEND * mv_writevars(SHORT chan, SHORT num_vars,
struct mv_var_desc *var_desc)
- wykonanie usługi Write przez odwołanie się do VMD
(por. funkcja mv_write);
SHORT chan - kanał, który posłuży do wykonania
usługi;
SHORT num_var - liczba zmiennych zapisywanych w usłudze
Write. Musi być ona równa liczbie struktur
typu mv_var_desc, które są wskazywane przez
var_desc;
struct mv_var_desc *var_desc - wskazanie na początek
tablicy struktur zawierających informacje o
zapisywanych zmiennych.
187. MMSREQ_PEND * mv_write_variables(SHORT chan,
SHORT num_data, struct var_acc_spec *vm_info,
struct mv_write_req_info *vm_data)
- wykonanie usługi Write przez odwołanie się do VMD
(por. funkcje mv_write oraz mv_writevars);
SHORT chan - kanał, który posłuży do wykonania
usługi;
SHORT num_data - liczba struktur w tablicy wskazywanej
przez vm_data;
struct var_acc_spec *vm_info - wskazanie na tablicę
struktur zawierających informacje o zmiennych
do zapisu;
struct mv_write_req_info *vm_data - wskazanie na
tablicę struktur zawierającą informację o
sposobie kodowania danych dla każdej zmiennej
branej pod uwagę w tej usłudze.
188. SHORT mv_cancel_resp(MMSREQ_IND *req_ptr,
SHORT success)
- przesłanie potwierdzenia na odpowiedź o wykonaniu
usługi Cancel przez odwołanie się do VMD;
MMSREQ_IND *req_ptr - wskazanie na strukturę
odpowiadającą pierwotnemu zleceniu
usługi (poprzedzającemu Cancel);
SHORT success = 0 - pozytywna odpowiedź dla Cancel i
odpowiedź o błędzie do usługi
kasowanej;
<> 0 - negatywna odpowiedź dla Cancel i
brak odpowiedzi do usługi
kasowanej.

189. SHORT mv_crepi_resp(struct MMSREQ_IND *req_ptr,
SHORT err_reason)
- przesłanie potwierdzenia na odpowiedź o wykonaniu
usługi CreateProgramInvocation przez odwołanie
się do VMD;
struct MMSREQ_IND *req_ptr - wskazanie na strukturę
przekazaną do funkcji u_crepi_ind.
SHORT err_reason - <> 0 -powod wysłania negatywnej
odpowiedzi.
190. SHORT mv_deftype_resp(struct MMSREQ_IND *indptr)
- przesłanie potwierdzenia na odpowiedź o wykonaniu
usługi DefineNamedType przez odwołanie się do VMD;
struct MMSREQ_IND *indptr - wskazanie na strukturę
uprzednio przekazaną w argumencie funkcji
u_deltype_ind.
191. SHORT mv_defvar_resp(struct MMSREQ_IND *indptr)
- przesłanie potwierdzenia na odpowiedź o wykonaniu
usługi DefinedNamedVariable przez odwołanie
się do VMD;
struct MMSREQ_IND *indptr - wskazanie na strukturę
uprzednio przekazaną w argumencie funkcji
u_defvar_ind.
192. SHORT mv_delpi_resp(struct MMSREQ_IND *req_ptr,
SHORT err_reason, SHORT soft)
- przesłanie potwierdzenia na odpowiedź o wykonaniu
usługi DeleteProgramInvocation przez odwołanie
się do VMD;
struct MMSREQ_IND *req_ptr - wskazanie na strukturę
przekazaną do funkcji u_delpi_ind.
SHORT err_reason - <> 0 -powod wysłania negatywnej
odpowiedzi;
SHORT soft - <> 0 - struktura prog_inv nie została
skasowana ale atrybut stanu są
umieszczono w klasie NON_EXISTENT,
= 0-struktura prog_inv została skasowana.
193. SHORT mv_deltype_resp(struct MMSREQ_IND *indptr)
- przesłanie potwierdzenia na odpowiedź o wykonaniu
usługi DeleteNamedType przez odwołanie się do VMD;
struct MMSREQ_IND *indptr - wskazanie na strukturę
uprzednio przekazaną w argumencie funkcji
u_delvar_ind.

194. SHORT mv_delvar_resp(struct MMSREQ_IND *indptr)
- przesłanie potwierdzenia na odpowiedź o wykonaniu usługi DeleteVariableAccess przez odwołanie się do VMD;
struct MMSREQ_IND *indptr - wskazanie na strukturę uprzednio przekazaną w argumencie funkcji u_delvar_ind.
195. SHORT mv_fclose_resp(struct MMSREQ_IND *req_ptr)
- przesłanie potwierdzenia na odpowiedź o wykonaniu usługi FileClose przez odwołanie się do VMD;
struct MMSREQ_IND *req_ptr - wskazanie na strukturę uprzednio przekazaną w argumencie funkcji u_fclose_ind.
196. SHORT mv_fdelete_resp(struct MMSREQ_IND *req_ptr)
- przesłanie potwierdzenia na odpowiedź o wykonaniu usługi FileDelete przez odwołanie się do VMD;
struct MMSREQ_IND *req_ptr - wskazanie na strukturę uprzednio przekazaną w argumencie funkcji u_fdelete_ind.
197. SHORT mv_fdir_resp(struct MMSREQ_IND *req_ptr)
- przesłanie potwierdzenia na odpowiedź o wykonaniu usługi FileDirectory przez odwołanie się do VMD;
struct MMSREQ_IND *req_ptr - wskazanie na strukturę uprzednio przekazaną w argumencie funkcji u_fdir_ind.
198. SHORT mv_fopen_resp(struct MMSREQ_IND *req_ptr)
- przesłanie potwierdzenia na odpowiedź o wykonaniu usługi FileOpen przez odwołanie się do VMD;
struct MMSREQ_IND *req_ptr - wskazanie na strukturę uprzednio przekazaną w argumencie funkcji u_fopen_ind.
199. SHORT mv_fread_resp(struct MMSREQ_IND *req_ptr)
- przesłanie potwierdzenia na odpowiedź o wykonaniu usługi FileRead przez odwołanie się do VMD;
struct MMSREQ_IND *req_ptr - wskazanie na strukturę uprzednio przekazaną w argumencie funkcji u_fread_ind.

200. SHORT mv_frename_resp(struct MMSREQ_IND *req_ptr)
 - przesłanie potwierdzenia na odpowiedź o wykonaniu usługi FileRename przez odwołanie się do VMD;
 struct MMSREQ_IND *req_ptr - wskazanie na strukturę uprzednio przekazaną w argumencie funkcji u_frename_ind.
201. SHORT mv_getpi_resp(struct MMSREQ_IND *req_ptr,
 SHORT err_reason)
 - przesłanie potwierdzenia na odpowiedź o wykonaniu usługi GetProgramInvocationAttributes przez odwołanie się do VMD;
 struct MMSREQ_IND *req_ptr - wskazanie na strukturę przekazaną do funkcji u_getpi_ind;
 SHORT err_reason - (<) 0 -powód wysłania negatywnej odpowiedzi.
202. SHORT mv_gettype_resp(struct MMSREQ_IND *req_ptr)
 - przesłanie potwierdzenia na odpowiedź o wykonaniu usługi GetNamedTypeAttributes przez odwołanie się do VMD;
 struct MMSREQ_IND *req_ptr - wskazanie na strukturę uprzednio przekazaną w argumencie funkcji u_gettype_ind.
203. SHORT mv_getvar_resp(struct MMSREQ_IND *req_ptr)
 - przesłanie potwierdzenia na odpowiedź o wykonaniu usługi GetVariableAccessAttributes przez odwołanie się do VMD;
 struct MMSREQ_IND *req_ptr - wskazanie na strukturę uprzednio przekazaną w argumencie funkcji u_getvar_ind.
204. SHORT mv_ident_resp(struct MMSREQ_IND *req_ptr)
 - przesłanie potwierdzenia na odpowiedź o wykonaniu usługi Identify przez odwołanie się do VMD;
 struct MMSREQ_IND *req_ptr - wskazanie na strukturę uprzednio przekazaną w argumencie funkcji u_ident_ind.
205. SHORT mv_init_resp(struct MMSREQ_IND *req_ptr)
 - przesłanie potwierdzenia na odpowiedź o wykonaniu usługi Initiate przez odwołanie się do VMD;
 struct MMSREQ_IND *req_ptr - wskazanie na strukturę uprzednio przekazaną w argumencie funkcji u_init_ind.

206. SHORT mv_kill_resp(struct MMSREQ_IND *req_ptr,
SHORT err_reason)
- przesłanie potwierdzenia na odpowiedź o wykonaniu
usługi Kill przez odwołanie się do VMD;
struct MMSREQ_IND *req_ptr - wskazanie na strukturę
przekazaną do funkcji u_kill_ind;
SHORT err_reason - <> 0 -powód wysłania negatywnej
odpowiedzi.
207. SHORT mv_namelist_resp(struct MMSREQ_IND *req_ptr)
- przesłanie potwierdzenia na odpowiedź o wykonaniu
usługi GetNamedList przez odwołanie się do VMD;
struct MMSREQ_IND *req_ptr - wskazanie na strukturę
uprzednio przekazaną w argumencie funkcji
u_namelist_ind.
208. SHORT mv_obtfile_resp(struct MMSREQ_IND *req_ptr)
- przesłanie potwierdzenia na odpowiedź o wykonaniu
usługi ObtainFile przez odwołanie się do VMD;
struct MMSREQ_IND *req_ptr - wskazanie na strukturę
uprzednio przekazaną w argumencie funkcji
u_obtfile_ind.
209. SHORT mv_read_resp(struct MMSREQ_IND *req_ptr)
- przesłanie potwierdzenia na odpowiedź o wykonaniu
usługi Read przez odwołanie się do VMD;
struct MMSREQ_IND *req_ptr - wskazanie na strukturę
uprzednio przekazaną w argumencie funkcji
u_read_ind.
210. SHORT mv_read_response(struct MMSREQ_IND *req_ptr,
SHORT num_var, struct mv_read_resp_info *respinfo)
- przesłanie potwierdzenia na odpowiedź o wykonaniu
usługi Read przez odwołanie się do VMD;
struct MMSREQ_IND *req_ptr - wskazanie na strukturę
uprzednio przekazaną w argumencie funkcji
u_read_ind;
SHORT num_var - liczba zmiennych czytanych w
odpowiedzi Read. Liczba ta musi być równa
liczbie struktur wskazywanych przez respinfo;
struct mv_read_resp_info *respinfo - wskazanie na
początek tablicy struktur zawierających
specyfikację poszczególnych czytanych
zmiennych.

211. SHORT mv_reset_resp(struct MMSREQ_IND *req_ptr,
SHORT err_reason)
- przesłanie potwierdzenia na odpowiedź o wykonaniu
usługi Reset przez odwołanie się do VMD;
struct MMSREQ_IND *req_ptr - wskazanie na strukturę
przekazaną do funkcji u_reset_ind.
SHORT err_reason - <> 0 -powód wysłania negatywnej
odpowiedzi.
212. SHORT mv_resume_resp(struct MMSREQ_IND *req_ptr,
SHORT err_reason)
- przesłanie potwierdzenia na odpowiedź o wykonaniu
usługi Resume przez odwołanie się do VMD;
struct MMSREQ_IND *req_ptr - wskazanie na strukturę
przekazaną do funkcji u_resume_ind.
SHORT err_reason - <> 0 -powód wysłania negatywnej
odpowiedzi.
213. SHORT mv_start_resp(struct MMSREQ_IND *req_ptr,
SHORT err_reason)
- przesłanie potwierdzenia na odpowiedź o wykonaniu
usługi Start przez odwołanie się do VMD;
struct MMSREQ_IND *req_ptr - wskazanie na strukturę
przekazaną do funkcji u_start_ind;
SHORT err_reason - <> 0 -powód wysłania negatywnej
odpowiedzi.
214. SHORT mv_stop_resp(struct MMSREQ_IND *req_ptr,
SHORT err_reason)
- przesłanie potwierdzenia na odpowiedź o wykonaniu
usługi Stop przez odwołanie się do VMD;
struct MMSREQ_IND *req_ptr - wskazanie na strukturę
przekazaną do funkcji u_stop_ind;
SHORT err_reason - <> 0 -powód wysłania negatywnej
odpowiedzi.
215. SHORT mv_write_resp(struct MMSREQ_IND *req_ptr)
- przesłanie potwierdzenia na odpowiedź o wykonaniu
usługi Write przez odwołanie się do VMD;
struct MMSREQ_IND *req_ptr - wskazanie na strukturę
uprzednio przekazaną w argumencie funkcji
u_write_ind.

216. VOID u_????_ind(MMSREQ_IND *req_ptr)

- jest to grupa funkcji wywoływana w momencie przyjscia potwierdzenia z oddalonego węzła sieci. Wszystkie one (poza omówionymi dalej, oddzielnie) mają przedstawiony wyżej format prototypu;

MMSREQ_IND *req_ptr - wskazanie na strukturę zawierającą informację o PDU otrzymanym w wyniku wykonania odpowiedniej, uprzednio wywołanej funkcji typu mp_???? (stanowi on również argument wspomnianej funkcji mp_????).

Poniżej zestawiono poszczególne funkcje typu u_????_ind i odpowiadające im zlecenia - usługi MMS:

u_abort_ind(SHORT,SHORT,SHORT) - Abort;
u_ackevnot_ind(MMSREQ_IND *) - AcknowledgeEventNotification;
u_altecm_ind(MMSREQ_IND *) - AlterEventConditionMonitoring;
u_altee_ind(MMSREQ_IND *) - AlterEventEnrollment;
u_cancel_ind(MMSREQ_IND *) - Cancel;
u_conclude_ind(MMSREQ_IND *) - Conclude;
u_crepi_ind(MMSREQ_IND *) - CreateProgramInvocation;
u_defea_ind(MMSREQ_IND *) - DefineEventAction;
u_defec_ind(MMSREQ_IND *) - DefineEventCondition;
u_defee_ind(MMSREQ_IND *) - DefineEventEnrollment;
u_defsem_ind(MMSREQ_IND *) - DefineSemaphore;
u_deftype_ind(MMSREQ_IND *) - DefineNamedType;
u_defvar_ind(MMSREQ_IND *) - DefineNamedVariable;
u_defvlist_ind(MMSREQ_IND *) - DefineNamedVariableList;
u_deldom_ind(MMSREQ_IND *) - DeleteDomain;
u_delea_ind(MMSREQ_IND *) - DeleteEventAction;
u_delec_ind(MMSREQ_IND *) - DeleteEventCondition;
u_delee_ind(MMSREQ_IND *) - DeleteEventEnrollment;
u_delpi_ind(MMSREQ_IND *) - DeleteProgramInvocation;
u_delsem_ind(MMSREQ_IND *) - DeleteSemaphore;
u_deltype_ind(MMSREQ_IND *) - DeleteNamedType;
u_delvar_ind(MMSREQ_IND *) - DeleteVariableAccess;
u_delvlist_ind(MMSREQ_IND *) - DeleteNamedVariableList;
u_download_ind(MMSREQ_IND *) - DownloadSegment;
u_evnot_ind(MMSREQ_IND *) - EventNotification;
u_fclose_ind(MMSREQ_IND *) - FileClose;
u_fdelete_ind(MMSREQ_IND *) - FileDelete;
u_fdir_ind(MMSREQ_IND *) - FileDirectory;
u_fopen_ind(MMSREQ_IND *) - FileOpen;

u_fread_ind(MMSREQ_IND *) - FileRead;
u_frename_ind(MMSREQ_IND *) - FileRename;
u_getaes_ind(MMSREQ_IND *) - GetAlarmEnrollmentSummary;
u_getas_ind(MMSREQ_IND *) - GetAlarmSummary;
u_getcl_ind(MMSREQ_IND *) - GetCapabilityList;
u_getdom_ind(MMSREQ_IND *) - GetDomainAttributes;
u_geteaa_ind(MMSREQ_IND *) - GetEventActionAttributes;
u_geteca_ind(MMSREQ_IND *) - GetEventConditionAttributes;
u_geteea_ind(MMSREQ_IND *) - GetEventEnrollmentAttributes;
u_getpi_ind(MMSREQ_IND *) - GetProgramInvocationAttributes;
u_gettype_ind(MMSREQ_IND *) - GetNamedTypeAttributes;
u_getvar_ind(MMSREQ_IND *) - GetVariableAccessAttributes;
u_getvlist_ind(MMSREQ_IND *) - GetNamedVariableListAttributes;
u_ident_ind(MMSREQ_IND *) - Identify;
u_initdown_ind(MMSREQ_IND *) - InitiateDownloadSequence;
u_initupl_ind(MMSREQ_IND *) - InitiateUploadSequence;
u_input_ind(MMSREQ_IND *) - Input;
u_jcreate_ind(MMSREQ_IND *) - CreateJournal;
u_jdelete_ind(MMSREQ_IND *) - DeleteJournal;
u_jinit_ind(MMSREQ_IND *) - InitializeJournal;
u_jread_ind(MMSREQ_IND *) - ReadJournal;
u_jstat_ind(MMSREQ_IND *) - ReportJournalStatus;
u_jwrite_ind(MMSREQ_IND *) - WriteJournal;
u_kill_ind(MMSREQ_IND *) - Kill;
u_llp_error_ind(SHORT, LONG);
u_loaddom_ind(MMSREQ_IND *) - LoadDomainContent;
u_mmsexcept_ind(SHORT, SHORT);
u_namelist_ind(MMSREQ_IND *) - GetNameList;
u_obtfile_ind(MMSREQ_IND *) - ObtainFile;
u_output_ind(MMSREQ_IND *) - Output;
u_rddwn_ind(MMSREQ_IND *) - RequestDomainDownload;
u_rdupl_ind(MMSREQ_IND *) - RequestDomainUpload;
u_read_ind(MMSREQ_IND *) - Read;
u_reject_ind(SHORT, struct reject_resp_info *);
u_relctrl_ind(MMSREQ_IND *) - RelinquishControl;
u_release_ind(SHORT) - Association Release;
u_rename_ind(MMSREQ_IND *) - Rename;
u_repeas_ind(MMSREQ_IND *) - ReportEventActionStatus;
u_repecs_ind(MMSREQ_IND *) - ReportEventConditionStatus;
u_repees_ind(MMSREQ_IND *) - ReportEventEnrollmentStatus;
u_reset_ind(MMSREQ_IND *) - Reset;
u_resume_ind(MMSREQ_IND *) - Resume;
u_rsenry_ind(MMSREQ_IND *) - ReportSemaphoreEntryStatus;
u_rspool_ind(MMSREQ_IND *) - ReportPoolSemaphoreStatus;
u_rsstat_ind(MMSREQ_IND *) - ReportSemaphoreStatus;
u_start_ind(MMSREQ_IND *) - Start;

- u_status_ind(MMSREQ_IND *) - Status;
 u_stop_ind(MMSREQ_IND *) - Stop;
 u_storedom_ind(MMSREQ_IND *) - StoreDomainContent;
 u_takectrl_ind(MMSREQ_IND *) - TakeControl;
 u_termdown_ind(MMSREQ_IND *) - TerminateDownloadSequence;
 u_termupl_ind(MMSREQ_IND *) - TerminateUploadSequence;
 u_trige_ind(MMSREQ_IND *) - TriggerEvent;
 u_upload_ind(MMSREQ_IND *) - UploadSegmentRequest;
 u_write_ind(MMSREQ_IND *) - Write;
217. VOID u_abort_ind(SHORT chan, SHORT reason, SHORT au_flag)
 SHORT chan - numer kanału z którego właśnie został
 otrzymany sygnał zaniechania;
 SHORT reason - kody zaniechania wg. specyfikacji MAP.
 SHORT au_flag - źródło zaniechania;
218. VOID u_cancel_ind(MMSREQ_IND *req_ptr)
 MMSREQ_IND *req_ptr - wskazanie na strukturę
 odpowiadającą kasowanemu zleceniu.
219. VOID u_conclude_ind(MMSREQ_PEND *req_ptr)
 MMSREQ_IND *req_ptr -wskazanie na strukturę
 zawierającą informacje otrzymane w zleceniu
 Conclude.
220. VOID u_llp_error_ind(SHORT chan, LONG errval)
 SHORT chan - numer kanału, w którym dostrzeżono błąd
 LLP.
 LONG errval - typ błędu LLP.
221. VOID u_mmsexcept_ind(SHORT chan, LONG errval)
 SHORT chan - numer kanału;
 LONG errval - wartość błędu MMS-EASE.
222. VOID u_reject_ind(SHORT chan,
 struct reject_resp_info *info_ptr)
 SHORT chan - numer kanału, w którym otrzymano Reject;
 struct reject_resp_info *info_ptr - wskazanie na