int openPort(const char \*fileName, long speed);

int closePort();

int checkPortAsync(const char \*fileName, long speed);

int commandStart();

MData libGetStatusFlags();

int getStatusFlags(int \*fatalStatus, int \*currentFlagsStatus, int \*documentStatus);

MData libGetCountersAndRegisters(unsigned char numRequest);

MData libGetKKTInfo(unsigned char numRequest);

int getKKTInfo(unsigned char numRequest, char \*data);

int getKKTInfoNum(unsigned char numRequest, int numParam, char \*param);

MData libGetReceiptData(unsigned char numRequest);

MData libGetPrinterStatus();

MData libGetServiceInfo(unsigned char numRequest);

MData libGetExErrorInfo(unsigned char numRequest);

int scrollPaper();

int libCommandStart(PLDate mpDate, PLTime mpTime);

MData libReadSettingsTable(unsigned char number, int index);

int libWriteSettingsTable(unsigned char number, int index, const char\* data);

MData libGetPiritDateTime();

int libSetPiritDateTime(PLDate mpDate, PLTime mpTime);

int libLoadLogo(int size, unsigned char\* data);

int libDeleteLogo();

int libLoadReceiptDesign(int size, unsigned char\* data);

int libLoadPicture(int width, int height, int sm, const char\* name, int number, unsigned char\* data);

int libPrintXReport(const char\* nameCashier);

int libPrintZReport(const char\* nameCashier, int options);

int libOpenDocument(unsigned char type, unsigned char numDepart, char\* nameCashier, long docNumber);

int libOpenDocumentEx(unsigned char type, unsigned char numDepart, char\* nameCashier, long docNumber, unsigned char taxN);

unsigned long libSetBuyerAddress(const char \*buyerAddress);

unsigned long libGetBuyerAddress(char \*buyerAddress, unsigned long baLength);

void libCleanBuyerAddress();

unsigned long libSetBuyerName(const char \*buyerName);

void libCleanBuyerName();

unsigned long libSetBuyerInn(const char \*buyerInn);

void libCleanBuyerInn();

unsigned long libSetCountryCode(const char \*countryCode);

unsigned long libSetDeclarationNum(const char \*declarationNum);

void libSetExciseSum(double exciseSum);

MData libCloseDocument(unsigned char cutPaper);

MData libCloseDocumentEx(unsigned char cutPaper, const char\* userTagName, const char\* userTagData);

int libCancelDocument();

int libPostponeDocument(const char\* info);

int libCutDocument();

int libPrintString(char\* textStr, unsigned char attribute);

int libPrintBarCode(unsigned char posText, unsigned char widthBarCode, unsigned char heightBarCode, unsigned char typeBarCode, const char\* barCode);

int libAddPosition(const char\* goodsName, const char\* barcode, double quantity, double price, unsigned char taxNumber, int numGoodsPos, unsigned char numDepart, unsigned char coefType, const char \*coefName, double coefValue);

int libAddPositionEx(const char\* goodsName, const char\* barcode, double quantity, double price, unsigned char taxNumber, int numGoodsPos, unsigned char numDepart, long signMethodCalculation, long signCalculationObject);

int libAddPositionLarge(const char\* goodsName, const char\* barcode, double quantity, double price, unsigned char taxNumber, int numGoodsPos, unsigned char numDepart, unsigned char coefType, double coefValue, long signMethodCalculation, long signCalculationObject);

int libDelPosition(const char\* goodsName, const char\* barcode, double quantity, double price, unsigned char taxNumber, int numGoodsPos, unsigned char numDepart);

int libSubTotal();

int libAddDiscount(unsigned char typeDiscount, const char\* nameDiscount, long sum);

int libAddMargin(unsigned char typeMargin, const char\* nameMargin, long sum);

int libAddPayment(unsigned char typePayment, long long sum, const char\* infoStr);

int libCashInOut(const char\* infoStr, long long sum);

int libPrintRequsit(unsigned char codeReq, unsigned char attributeText, const char\* str1, const char\* str2, const char\* str3, const char\* str4);

int libPrintRequsitOFD(int codeReq, unsigned char attributeText, const char\* reqName, const char\* reqStr);

int libRegisterSumToDepart(unsigned char typeOperation, unsigned char numberDepart, long sum);

int libRegisterTaxSum(unsigned char numberTax, long sum);

int libCompareSum(long sum);

int libOpenCopyReceipt(unsigned char type, unsigned char numDepart, const char\* nameCashier, int numCheck, int numCash, PLDate mpDate, PLTime mpTime);

int libOpenCopyReceiptEx(unsigned char type, unsigned char numDepart, const char\* nameCashier, int numCheck, int numCash, int docNum, int taxN);

int libSetToZeroCashInCashDrawer();

int libPrintPictureInDocument(int width, int height, int sm, unsigned char\* data);

int libPrintPreloadedPicture(int sm, int number);

int libTechnologicalReset(const PLDateTime \*dateTime);

int libFiscalization(const char \*oldPassword, const char \*regNumber, const char \*INN, const char \*newPassword);

int libPrintFiscalReportByShifts(unsigned char typeReport, int startShiftNumber, int endShiftNumber, const char \*password);

int libPrintFiscalReportByDate(unsigned char typeReport, PLDate startDate, PLDate endDate, const char \*password);

int libActivizationECT();

int libCloseArchiveECT();

int libCloseFN(const char \*cashierName);

int libPrintControlTapeFromECT(int shiftNumber);

int libPrintDocumentFromECT(int KPKNumber);

int libPrintReportFromECTByShifts(unsigned char typeReport, int startShiftNumber, int endShiftNumber);

int libPrintReportFromECTByDate(unsigned char typeReport, PLDate startDate, PLDate endDate);

int libPrintReportActivizationECT();

int libPrintReportFromECTByShift(int shiftNumber);

MData libGetInfoFromECT(unsigned char number, long dataL1, long dataL2);

int libOpenCashDrawer(int pulseDuration);

MData libGetCashDrawerStatus();

int getCashDrawerStatus(int \*drawerStatus);

int libBeep(int duration);

int libAuthorization(PLDate mpDate, PLTime mpTime, const char \*numKKT);

MData libReadMemoryBlock(unsigned char type, long startAdress, long numBytes);

int libSetSpeed(unsigned char numSpeed);

int libPrintServiceData();

int libEmergencyCloseShift();

int libPrintCopyLastZReport();

int libEnableServiceChannelToECT();

int libPrintCopyReportFiscalization();

int getCurMPTime(PLDate \*mpDate, PLTime \*mpTime);

int saveLogoToFile(wchar\_t \*fileName);

void setDebugLevel(int level);

long long getDriverVersion();

int libPrintDocsFromECTSDByNumberDoc(int startNumber, int endNumber);

int libPrintDocsFromECTSDByNumberShift(int startNumber, int endNumber);

int libPrintDocsFromECTSDByDate(PLDate mpDateStart, PLDate mpDateEnd);

int libGetInfoFromECT\_NumberDoc(int \*numDoc);

int libGetInfoFromECT\_ShiftNumber(int \*shiftNumber);

int libBLRPrintControlTapeFromECT();

int getCountersAndRegisters(int requestNumber, int \*data);

void setAmountDecimalPlaces(int decimalPlaces);

void setQuantityDecimalPlaces(int decimalPlaces);

void setPercentageDecimalPlaces(int decimalPlaces);

int getAmountDecimalPlaces();

int getQuantityDecimalPlaces();

int getPercentageDecimalPlaces();

int libGetCountersAndRegisters\_12(int data[], int maxElement, char \*str);

int getPrinterStatus(int\* result);

int getPiritDateTime(int\* cDate, int\* cTime);

int libOpenShift(const char\* nameCashier);

int libSetExtraRequisite(const char\* nomenclatureCode, const char\* extReq, const char\* measureName, int agentSign, const char\* supplierINN, const char\* supplierPhone, const char\* supplierName, const char\* operatorAddress, const char\* operatorINN, const char\* operatorName, const char\* operatorPhone, const char\* payAgentOperation, const char\* payAgentPhone, const char\* recOperatorPhone);

int libDoCheckCorrection(const char \*nameCashier, double cash, double cashless, unsigned char correctionFlags);

int libDoCheckCorrectionEx(const char \*nameCashier, double cash, double cashless, double sum1, double sum2, double sum3, unsigned char correctionFlags, const PLDate \*docDate, const char \*docNumber, const char \*correctionName, double sum\_18, double sum\_10, double sum\_0, double sum\_WT, double sum\_18\_118, double sum\_10\_110);

int libDoCheckCorrectionExTag(const char \*nameCashier, double cash, double cashless, double sum1, double sum2, double sum3, unsigned char correctionFlags, const PLDate \*docDate, const char \*docNumber, const char \*correctionName, double sum\_18, double sum\_10, double sum\_0, double sum\_WT, double sum\_18\_118, double sum\_10\_110, const char \*extraTag);

int libCurrentStatusReport(char\* nameCashier);

int libAddPaymentD(unsigned char typePayment, double sum, const char\* infoStr);

int libRegistration(unsigned char type, const char \*regNumber, const char \*INN, int systemTax, int rej, const char \*cashierName);

int libRegistrationEx(unsigned char type, const char \*regNumber, const char \*INN, int systemTax, int rej, const char \*cashierName, PLDateTime regDateTime, const char \*userName1, const char \*userName2, const char \*addressSettle1, const char \*addressSettle2, const char \*placeSettle, const char \*autoNumber, const char \*OFDName, const char \*OFDINN, const char \*senderEmail, const char \*FNSEmail, bool automatic);

int libGetInfoFromECT\_NumberFP(char \*data);

int libGetInfoFromECT\_FP(int numDoc, char \*data);

unsigned long libFormatMessage(int errorCode, char \*msgBuffer, unsigned long cbBuffer);

MData libGetPaidServiceStatus(int code);

MData libSendCommand(piritCommand\_t cmd, int num, ...);

int libSendCommandAsync(piritCommand\_t cmd, int num, ...);