

Федеральное государственное унитарное предприятие
"РОССИЙСКИЙ ФЕДЕРАЛЬНЫЙ ЯДЕРНЫЙ ЦЕНТР -
Всероссийский научно-исследовательский институт
технической физики имени академика Е.И. Забабахина"
(ФГУП «РФЯЦ-ВНИИТФ им. академ. Е.И. Забабахина»)

Система автоматизированной поверки и калибровки "Сапсан"

Руководство программиста

Содержание

1. НАЗНАЧЕНИЕ И УСЛОВИЯ ПРИМЕНЕНИЯ	3
2. РЕДАКТОР LUA.....	4
3. ОПИСАНИЕ КОМАНД.....	6
3.1. Общие команды.....	6
3.2. Команды пользовательского интерфейса	8
3.3. Команды формирования протокола.....	12
3.4. Команды стандарта VISA	14
3.5. Команды управления калибратором Fluke 9500B	16
4. ТЕРМИНЫ И СОКРАЩЕНИЯ.....	19

1. НАЗНАЧЕНИЕ И УСЛОВИЯ ПРИМЕНЕНИЯ

ПО должно стать инструментом информационной поддержки организационно-управленческих процессов, эффективного управления и использования информационных ресурсов, формирования среды для профессионального метрологического обслуживания средств измерений.

Основными целями создания ПО являются:

- формирование единого информационного пространства и предоставление необходимой информации в соответствии с потребностями отдельных пользователей и групп пользователей;
- автоматизация процесса поверки и калибровки;
- снижение влияния человеческого фактора на результаты измерений и расчетов;
- сокращение времени работы эталонных средств измерений;
- экономия времени при обработке результатов измерений.

Системные требования к персональному компьютеру для работы с ПО:

- операционная система Microsoft Windows 7 и выше;
- оперативная память 2 ГБ и выше;
- процессор Intel Core i3 и выше;
- видеокарта GeForce GT 440 и выше;
- интерфейсы обмена данными GPIB, USB, RS232, Ethernet.

Требования к квалификации программиста:

- высшее или среднее специальное образование;
- технические знания в написании скрипт-процедур на языке Lua;
- технические знания стандарта VISA (при организации связи с приборами поддерживающими стандарт VISA);
- технические знания в написании C-библиотек для Lua (при организации связи с приборами не поддерживающими стандарт VISA).

2. РЕДАКТОР LUA

Активация вкладки «Редактор Lua» (рис. 1) происходит из вкладки «Процедуры» при добавлении или редактировании процедуры. После установки ПО содержит примеры процедур, их наименования начинаются со слова «Example». Правила работы со вкладкой «Процедуры» описано в «Система автоматизированной поверки и калибровки "Сапсан". Руководство пользователя».

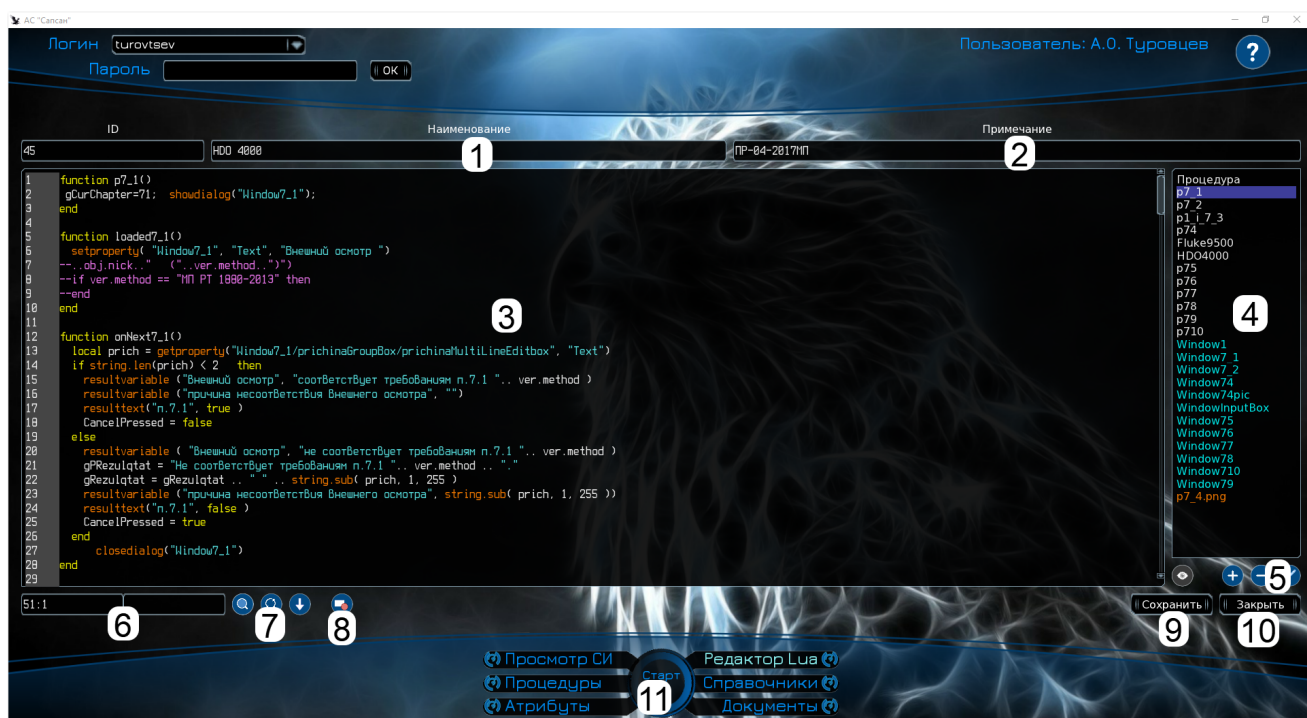


Рисунок 1 – Вкладка «Редактор Lua»

В расположенных сверху полях (поз. 1 и 2 рис. 1) вводятся наименование и примечание процедуры. В текстовом поле (поз. 3 рис. 1) вводится текст процедуры или XML описание пользовательского окна в зависимости от выбора в списке (поз. 4 рис. 1). Кнопки (поз. 5 рис. 1) под этим списком предназначены для добавления, удаления и переименования модулей, пользовательских окон, рисунков. Кнопка «Сохранить» (поз. 9 рис. 1) предназначена для сохранения внесенных изменений. Кнопка «Закрыть» (поз. 10 рис. 1) предназначена для закрытия редактируемой процедуры. В расположенных снизу полях (поз. 6 рис. 1) отображается текущее положение курсора в текстовом поле и признак наличия несохраненных изменений в редактируемой процедуре. Кнопки (поз. 7 рис. 1) предназначены для поиска, замены и переходу к строке с определенным номером. Кнопка (поз. 8 рис. 1) предназначена для установки и снятия точек останова для режима отладки.

Текст процедуры пишется на языке Lua версии 5.3, официальное справочное руководство по языку Lua 5.3 находится на сайте <http://www.lua.org/manual/5.3/>. Пользовательские окна создаются при помощи Crazy Eddie's GUI System, справочное руководство по XML описанию окон находится

на официальном сайте <http://www.cegui.org.uk>. Также на этом сайте можно скачать дизайнер окон CEED, который автоматически формирует XML описание созданного окна.

Для запуска текущей процедуры необходимо нажать кнопку «Старт» (поз. 11 рис. 1) или клавишу F5. При этом ПО переключается в режим отладки (рис. 2). В левом верхнем списке (поз. 4 рис. 2) отображается информация о ходе выполнения процедуры. В правом верхнем списке (поз. 5 рис. 2) отображаются текущие значения переменных, отсортированные по наименованию. Если необходимо отслеживать значение определенной переменной, то можно перенести ее в начало списка кликнув два раза левой кнопкой мыши по этой переменной. В левом нижнем списке (поз. 6 рис. 2) отображается передаваемая и получаемая по стандарту VISA информация. В правом нижнем списке (поз. 7 рис. 2) отображается исходный текст процедуры с выделением следующей выполняемой строки. В текстовом поле (поз. 1 рис. 2) отображается наименование модуля с номером следующей выполняемой строки. Кнопка «Шаг» (поз. 3 рис. 2) предназначена для выполнения одной строки кода процедуры. Кнопка «Старт» (поз. 2 рис. 2) предназначена для запуска процедуры до завершения или нажатия кнопки «Пауза» (появляется вместо кнопки «Старт» после запуска). Кнопка «Стоп» (поз. 8 рис. 2) позволяет остановить выполнение процедуры.



Рисунок 2 – Режим отладки

3. ОПИСАНИЕ КОМАНД

3.1. Общие команды

sleep(milliseconds)

приостановить выполнение программы на заданное количество миллисекунд

milliseconds количество миллисекунд, тип integer

savestate(state)

сохранить строку данных для возможности продолжения процедуры после прерывания (для сохранения нескольких переменных можно использовать string.pack)

state сохраняемое значение, тип string

Примечание: после полного завершения процедуры необходимо сохранять пустую строку, иначе процедура будет иметь флаг незавершенности

loadstate()

загрузить строку данных, сохраненную при предыдущем запуске процедуры

starttimer(delay)

запустить таймер, который будет периодически вызывать функцию onTimer()

delay время срабатывания в миллисекундах, тип integer

stoptimer()

остановить таймер

devicegetinfo(info)

возвращает информацию о проверяемом СИ из БД

info тип возвращаемой информации:

- "name" – наименование СИ
- "type" – тип СИ
- "serial" – заводской номер
- "owner" – владелец
- "manufacturer" – изготовитель
- "function" – назначение ("рабочее", "средство поверки")

"fif" – номер в Федеральном Информационном Фонде

"mpi" – межповерочный интервал

"year" – год изготовления

"reg_number" – регистрационный номер эталона

devicegetattribute(attribute, name)

возвращает атрибут поверяемого СИ (при отсутствии выбранного атрибута будет возвращен nil)

attribute атрибут, тип string

name наименование из списка (оставлять пустым при отсутствии списка), тип string

calibratorslist()

возвращает список всех средств поверки

Пример:

```
calibs = calibratorslist()
```

```
for j = 1, #calibs do
```

```
    local n = calibratorgetinfo(calibs[j].manufacturer, calibs[j].serial, "name")
```

```
    local t = calibratorgetinfo(calibs[j].manufacturer, calibs[j].serial, "type")
```

```
    resulttext(n.." "..t.." "..calibs[j].serial)
```

```
end
```

calibratorgetinfo(manufacturer, serial, info)

возвращает информацию о выбранном средстве поверки из БД (при отсутствии выбранного средства поверки будет возвращен nil)

manufacturer изготовитель, тип string

serial заводской номер, тип string

info тип возвращаемой информации:

"name" – наименование СИ

"type" – тип СИ

"serial" – заводской номер

"owner" – владелец

"manufacturer" – изготовитель

"function" – назначение ("рабочее", "средство поверки")

"fif" – номер в Федеральном Информационном Фонде

"mpi" – межповерочный интервал

"year" – год изготовления

"reg_number" – регистрационный номер эталона

calibratorgetattribute(manufacturer, serial, attribute, name)

возвращает атрибут выбранного средства поверки (при отсутствии выбранного средства поверки или атрибута будет возвращен nil)

manufacturer изготовитель, тип string

serial заводской номер, тип string

attribute атрибут, тип string

name наименование из списка (оставлять пустым при отсутствии списка), тип string

3.2. Команды пользовательского интерфейса

messagebox(text, caption[, type])

вывести окно с сообщением, возвращает 1 при нажатии в окне кнопки «Да» или «ОК» и 0 в остальных случаях


text текст сообщения, тип string

caption заголовок окна, тип string

type тип окна, формируется из суммы необходимых флагов (по умолчанию вызывается окно без иконок с кнопкой «ОК»):

MB_YESNO – окно с кнопками «Да» «Нет»

MB_OKCANCEL – окно с кнопками «ОК» «Отмена» (при отсутствии этих флагов будет окно с кнопкой «ОК»)

MB_ICONERROR – окно с иконкой 

MB_ICONQUESTION – окно с иконкой 

MB_ICONWARNING – окно с иконкой 

MB_ICONINFORMATION – окно с иконкой 

inputbox(text, caption)

вывести окно с запросом ввода, возвращает введенную пользователем строку или `nil`, если пользователь нажал «Отмена»

text текст сообщения, тип `string`

caption заголовок окна, тип `string`

getlastkeydown()

возвращает код последней нажатой клавиши

showdialog(window)

показать диалоговое окно

window наименование окна, тип `string`

closedialog(window)

закрыть диалоговое окно

window наименование окна, тип `string`

isdialogpresent(window)

возвращает `true`, если заданное диалоговое окно открыто

window наименование окна, тип `string`

getproperty(name, property)

возвращает выбранное свойство элемента заданного диалогового окна

name наименование элемента окна, тип `string`

property наименование свойства, тип `string`

setproperty(name, property, value)

установить выбранное свойство элемента заданного диалогового окна

name наименование элемента окна, тип `string`

property наименование свойства, тип `string`

value устанавливаемое значение, тип `string`

activate(name)

установить фокус ввода на элемент заданного диалогового окна

name наименование элемента окна, тип `string`

resetlistbox(name)

очистить listbox

name наименование listbox'а, тип string

getlistboxitemcount(name)

возвращает количество записей в listbox'е

name наименование listbox'а, тип string

addlistboxitem(name, value, id)

добавить запись в listbox

name наименование listbox'а, тип string

value добавляемая запись, тип string

id идентификатор добавляемой записи, тип integer

dellistboxitem(name, index)

удалить запись из listbox'а

name наименование listbox'а, тип string

index индекс удаляемой записи, тип integer

getlistboxitem(name, index)

возвращает идентификатор и значение записи в listbox'е

name наименование listbox'а, тип string

index индекс записи (допускается использовать “-1” для выделенной записи), тип integer

setlistboxitemselect(name, index)

выделить запись в listbox'е

name наименование listbox'а, тип string

index индекс выделяемой записи, тип integer

resetcombobox(name)

очистить combobox

name наименование combobox'а, тип string

addcomboboxitem(name, value, id)

добавить запись в combobox

<i>name</i>	наименование combobox'а, тип string
<i>value</i>	добавляемая запись, тип string
<i>id</i>	идентификатор добавляемой записи, тип integer

getcomboboxitem(name, index)

возвращает идентификатор и значение записи в combobox'е

<i>name</i>	наименование combobox'а, тип string
<i>index</i>	индекс записи (допускается использовать “-1” для выбранной записи), тип integer

resetmultilist(name)

очистить multilist

<i>name</i>	наименование multilist'а, тип string
-------------	--------------------------------------

getmultilistrowcount(name)

возвращает количество строк в multilist'е

<i>name</i>	наименование multilist'а, тип string
-------------	--------------------------------------

addmultilistrows(name, count)

добавить строки в multilist

<i>name</i>	наименование multilist'а, тип string
<i>count</i>	количество строк, тип integer

delmultilistrow(name, row)

удалить строку из multilist'а

<i>name</i>	наименование multilist'а, тип string
<i>row</i>	номер строки, тип integer

setmultilistitem(name, value, id, column, row)

установить значение ячейки в multilist'е

<i>name</i>	наименование multilist'а, тип string
<i>value</i>	новое значение, тип string
<i>id</i>	новый идентификатор, тип integer

<i>column</i>	номер столбца, тип integer
<i>row</i>	номер строки (допускается использовать “-1” для выделенной ячейки), тип integer

getmultilistitem(name, column, row)

возвращает идентификатор и значение ячейки в multilist’е

<i>name</i>	наименование multilist’а, тип string
<i>column</i>	номер столбца, тип integer
<i>row</i>	номер строки (допускается использовать “-1” для выделенной ячейки), тип integer

setmultilistitemselect(name, column, row)

устанавливает выделенную ячейку в multilist’е

<i>name</i>	наименование multilist’а, тип string
<i>column</i>	номер столбца, тип integer
<i>row</i>	номер строки, тип integer

getmultilistitemselect(name)

возвращает номер столбца и строки выделенной ячейки в multilist’е

<i>name</i>	наименование multilist’а, тип string
-------------	--------------------------------------

3.3. Команды формирования протокола

resulttext(text[, result[, var_name]])

записать текстовое значение в протокол

<i>text</i>	текстовое значение, тип string
<i>result</i>	результат записываемого значения (при отсутствии этого параметра текстовое значение будет считаться заголовком): true – годеи false – брак
<i>var_name</i>	наименование пользовательской переменной, тип string (при наличии этого параметра измеренное значение будет скопировано в пользовательские переменные)

resultvalue(set, real, min, max, unit, precision[, var_name])

записать числовое значение в протокол, возвращает 1 при $\min \leq \text{real} \leq \max$ и 0 в остальных случаях

<i>set</i>	заданное значение измеряемой величины, тип number (допускается использовать nil при отсутствии заданного значения)
<i>real</i>	измеренное значение, тип number
<i>min</i>	минимально допустимое значение, тип number (допускается использовать nil при отсутствии минимальной границы для измеренного значения)
<i>max</i>	максимально допустимое значение, тип number (допускается использовать nil при отсутствии максимальной границы для измеренного значения)
<i>unit</i>	единица измерения, тип string
<i>precision</i>	число значащих цифр после запятой, тип integer
<i>var_name</i>	наименование пользовательской переменной, тип string (при наличии этого параметра измеренное значение будет скопировано в пользовательские переменные)

resultvariable(var_name, value)

записать пользовательскую или системную переменную в протокол

<i>var_name</i>	наименование переменной, тип string для системных переменных возможны значения: "temperature" – температура (°C) "humidity" – влажность (%) "pressure" – давление (мм рт. ст.) "power_voltage" – питание сети (В) "power_frequency" – питание сети (Гц) "calibration" – вид МО
<i>value</i>	значение переменной, тип string для calibration возможные значения: «первичная поверка», «периодическая поверка», «калибровка»

getvariable(var_name)

возвращает пользовательскую переменную из протокола

<i>var_name</i>	наименование переменной, тип string
-----------------	-------------------------------------

Примечание: если процедура запущена не из вкладки «Просмотр СИ», то возвращает пустую строку

resulttable(tbl_idx, column, row, value)

записать значение в пользовательскую таблицу

<i>tbl_idx</i>	номер таблицы, тип integer (от 1 до 255)
<i>column</i>	номер столбца, тип integer (от 1 до 255)
<i>row</i>	номер строки, тип integer (от 1 до 65535)
<i>value</i>	записываемое значение, тип string (не более 255 символов)

getresult()

возвращает результат всех записанных значений: “годен” или “брак” (при отсутствии записанных значений возвращает пустую строку)

setresult(result)

устанавливает значение, отображаемое в поле «Заключение» окна «Сохранить протокол»

<i>result</i>	заключение: «годен» или «брак»
---------------	--------------------------------

calibratoradd(manufacturer, serial)

добавляет средство поверки в протокол, возвращает 1 при успешном добавлении и 0 при отсутствии такого средства поверки в БД

<i>manufacturer</i>	изготовитель, тип string
<i>serial</i>	заводской номер, тип string

Примечание: обязательно наличие действующего свидетельства или сертификата у данного средства поверки

3.4. Команды стандарта VISA

visafindrsrc(name)

возвращает таблицу со списком символьных имен найденных ресурсов visa

<i>name</i>	строка для поиска ресурса, тип string
	допускается использование специальных символов:
	? соответствует одному любому символу
	* соответствует любому количеству повторений предыдущего символа
	[<i>список</i>] соответствует одному любому символу из списка

visawrite(name, buf[, size])

записать данные в прибор

<i>name</i>	символьное имя ресурса visa, тип string
<i>buf</i>	данные для записи, тип string
<i>size</i>	количество отправляемых байт если в buf двоичные данные, а не строковые, тип integer

visaread(name[, timeout[, size]])

считать данные с прибора

<i>name</i>	символьное имя ресурса visa, тип string
<i>timeout</i>	время ожидания ответа от прибора в миллисекундах (по умолчанию 2000 мс, допускается устанавливать “-1” для бесконечного ожидания), тип integer
<i>size</i>	размер буфера для считываемых данных в байтах (по умолчанию 1024 байт), тип integer

visasetattribute(name, attribute, state)

изменить значение атрибута

<i>name</i>	символьное имя ресурса visa, тип string
<i>attribute</i>	атрибут, тип integer
	VI_ATTR_TMO_VALUE – таймаут ожидания ответа (по умолчанию 2000 мс)
	VI_ATTR_ASRL_BAUD – скорость порта (по умолчанию 9600)
	VI_ATTR_ASRL_DATA_BITS – биты данных (по умолчанию 8)
	VI_ATTR_ASRL_PARITY – четность (по умолчанию VI_ASRL_PAR_NONE)
	VI_ATTR_ASRL_STOP_BITS – стоп биты (по умолчанию VI_ASRL_STOP_ONE)
	VI_ATTR_ASRL_FLOW_CNTRL – управление потоком (по умолчанию VI_ASRL_FLOW_NONE)
	VI_ATTR_TERMCHAR - завершающий символ (по умолчанию 0Ah)
	VI_ATTR_TERMCHAR_EN - флаг разрешения использования завершающего символа (по умолчанию 0)
	VI_ATTR_ASRL_RTS_STATE - состояние сигнала Request To Send
	VI_ATTR_ASRL_DTR_STATE - состояние сигнала Data Terminal Ready
	VI_ATTR_ASRL_END_IN - метод прерывания операции чтения (по умолчанию VI_ASRL_END_TERMCHAR)

state значение атрибута, тип integer

для VI_ATTR_ASRL_PARITY возможные значения: VI_ASRL_PAR_NONE, VI_ASRL_PAR_ODD, VI_ASRL_PAR_EVEN, VI_ASRL_PAR_MARK, VI_ASRL_PAR_SPACE

для VI_ATTR_ASRL_STOP_BITS возможные значения: VI_ASRL_STOP_ONE, VI_ASRL_STOP_ONE5, VI_ASRL_STOP_TWO

для VI_ATTR_ASRL_FLOW_CNTRL возможные значения: VI_ASRL_FLOW_NONE, VI_ASRL_FLOW_XON_XOFF, VI_ASRL_FLOW_RTS_CTS, VI_ASRL_FLOW_DTR_DSR

для VI_ATTR_ASRL_END_IN возможные значения: VI_ASRL_END_NONE, VI_ASRL_END_LAST_BIT, VI_ASRL_END_TERMCHAR, VI_ASRL_END_BREAK

visaclose()

закрывает все открытые подключения

Примечание: вызывается автоматически при окончании процедуры

3.5. Команды управления калибратором Fluke 9500B

fluke9500dc(level)

установить выход калибратора в режим постоянного напряжения

level напряжение в вольтах

fluke9500sine(level, frequency)

установить выход калибратора в режим синусоидального сигнала

level амплитуда сигнала в вольтах

frequency частота сигнала в герцах

fluke9500square(level, frequency, polarity)

установить выход калибратора в режим сигнала прямоугольной формы

level амплитуда сигнала в вольтах

frequency частота сигнала в герцах

polarity полярность сигнала:

“pos” – сигнал положительной полярности относительно земли

“neg” – сигнал отрицательной полярности относительно земли

“symm” – симметричный сигнал относительно земли

fluke9500edge(level, frequency, speed)

установить выход калибратора в режим функции перепада импульса

<i>level</i>	амплитуда сигнала в вольтах
<i>frequency</i>	частота сигнала в герцах
<i>speed</i>	скорость нарастания выходного напряжения в секундах (калибратор поддерживает 150 пс «Fast Edge», 500 пс или 100 нс «High Edge»)

fluke9500marker(form, level, period)

установить выход калибратора в режим сигнала временного маркера

<i>form</i>	форма сигнала: “square” или “sine” – прямоугольный/синусоидальный сигнал “pulse” – импульсный сигнал “triangle” – узкий треугольный сигнал “line” – прямоугольный сигнал с частотой сетевого напряжения
<i>level</i>	амплитуда сигнала в вольтах
<i>period</i>	период сигнала в секундах

fluke9500resistance(channel, impedance)

возвращает результат функции измерения входного сопротивления испытуемого осциллографа

<i>channel</i>	канал, используемый для измерения: “1”, “2”, “3”, “4” или “5”
<i>impedance</i>	ожидаемый импеданс нагрузки выбранного сигнального канала в Омах

fluke9500trigger(channel)

выбрать канал, связанный с выходом запуска

<i>channel</i>	канал, связанный с выходом запуска: “0”, “1”, “2”, “3”, “4” или “5” (“0” - отключает канал запуска)
----------------	---

fluke9500head(channel)

возвращает тип, серийный номер, дату последней калибровки и дату следующей калибровки активной головки, подключенной к выбранному каналу

channel канал: “1”, “2”, “3”, “4” или “5”

fluke9500on(channel, impedance)

включить выхода калибратора

channel канал, используемый для выходного сигнала: “1”, “2”, “3”, “4” или “5”

impedance ожидаемый импеданс нагрузки выбранного сигнального канала в Омах

fluke9500off()

выключить выход калибратора

4. ТЕРМИНЫ И СОКРАЩЕНИЯ

Термин	Полная форма
БД	база данных
ПО	программное обеспечение
СИ	средство измерения

