

- секция 2 – передача результата АЦП и признака превышения порога через UART во внешнее устройство;
- секция 3 – приём значения порога превышения от внешнего устройства через UART.

Периодичность передачи управления внутрь первых двух секций задаётся таймерами 0 и 1. Третья секция выполняется асинхронно, по факту поступления данных от внешнего устройства.

РАЗРАБОТКА УПРАВЛЯЮЩЕЙ ПРОГРАММЫ

Для написания встроенной управляющей программы графического процессора необходимо в Workshop4 IDE создать ViSi-проект, предварительно подключив дисплейный модуль к USB-порту компьютера через адаптер μ USB-PA5. После обнаружения ОС Windows USB-подключения требуется уточнить номер созданного в системе виртуального COM-порта.

Для создания ViSi-проекта в главном окне Workshop4 IDE следует выбрать Create a new Project, после чего задать целевой дисплейный модуль (в нашем случае – μ OLED-128-G2), как показано на рисунке 6, и нажать кнопку Next. Затем в открывшемся окне выбора вида проекта нажать кнопку со стрелкой около иконки ViSi, как показано на рисунке 7.

В окне редактора в центре располагается созданный по умолчанию шаблон исходного кода программы (Program Skeleton) на языке 4DGL, справа сверху – поле дисплея выбранного модуля, справа снизу – область диспетчера объектов проекта (см. рис. 8). Созданный ViSi-проект рекомендуется сразу же сохранить в отдельном каталоге.

Номер COM-порта выбирается из выпадающего списка в меню Comms (см. рис. 9). При установлении соединения между компьютером и модулем кружок в поле Comms сменит цвет с красного сначала на жёлтый, а затем – на синий. Здесь же появится информация о подключённом модуле и карте памяти (при наличии последней). Однако вставлять карту памяти в модуль на данном этапе не следует, так как на неё будет производиться первоначальная запись медийных объектов. Если планируется использовать в приложении медийные функции, осуществлять вывод на дисплей сложных графических объектов, изображений, воспроизводить видео, отображать текстовую информацию пользовательским шрифтом, то их

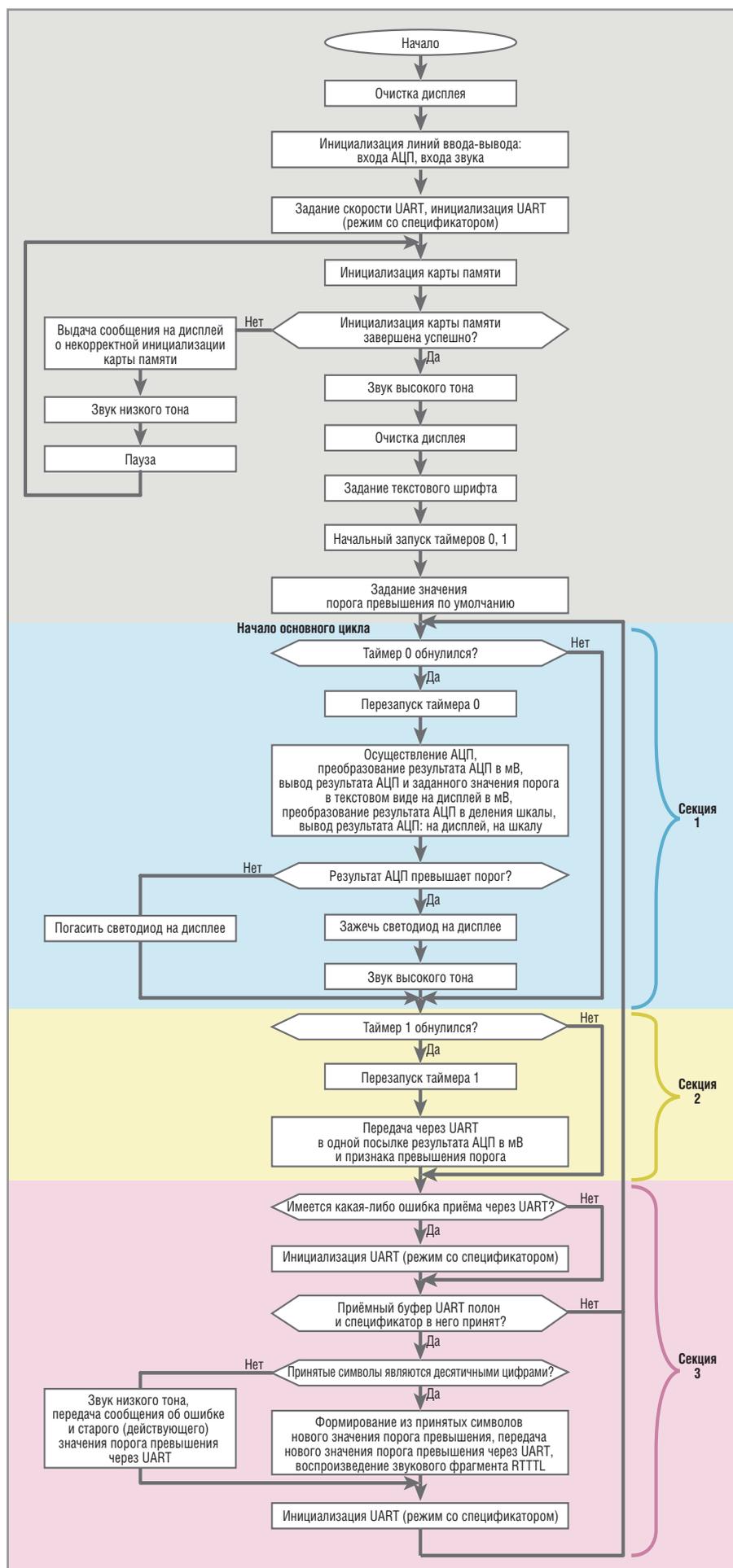


Рис. 5. Блок-схема алгоритма встроенной управляющей программы



Рис. 6. Задание целевого дисплейного модуля при создании проекта



Рис. 7. Выбор вида проекта – ViSi-проект

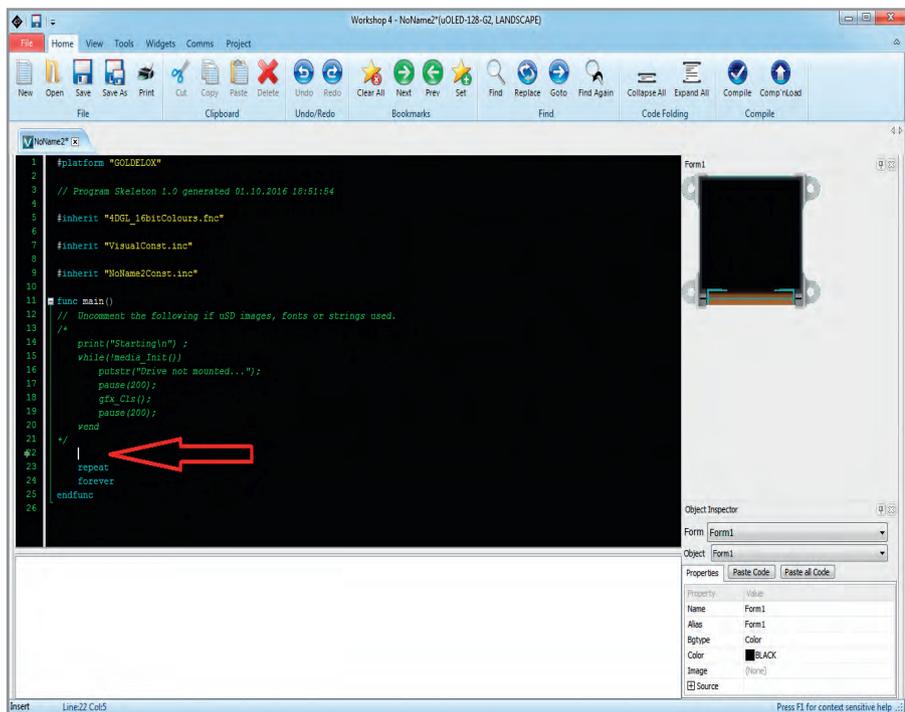


Рис. 8. Окно редактора с шаблоном исходного кода ViSi-проекта

следует предварительно записать на µSD-карту, воспользовавшись, например, кардридером. На этом подготовку аппаратной части к созданию встраиваемого приложения можно считать завершённой.

Как можно видеть из рисунка 8, в шаблоне исходного кода имеется закомментированный программный блок, осуществляющий начальную инициализацию карты памяти. В случае использования карты его необходимо раскомментировать. В данном варианте программы в этот блок, наряду с командами вывода на дисплей текстовых сообщений, добавлены команды генерации звука (см. рис. 5). Звукom высокого тона пользователь оповещается об успешной инициализации µSD-карты и передаче управления в следующий блок программы, низкого – об

ошибке инициализации или отсутствии карты в слоте модуля, после чего следует новая попытка инициализации.

Блоку инициализации карты памяти в программе предшествует блок инициализации UART и линий GPIO: IO1 сконфигурирован как аналоговый вход, а IO2 – как выход звука.

Основная функция программы, располагается в шаблоне между командами `func main()` и `endfunc`. Заполнить шаблон программным кодом разработчику предстоит вручную, за исключением блоков описания медийных объектов, которые создаются Workshop4 IDE автоматически. Медийных объектов в нашем приложении будет три: шрифт, графическая шкала и символ «светодиода». Начнём их создание со шрифта.



Рис. 9. Установленное соединение между компьютером и графическим модулем

В шаблоне исходного кода следует установить курсор в позицию, отмеченную на рисунке 8 красной стрелкой. Необходимо отметить, что инициализация объекта в коде всегда должна предшествовать первому обращению программы к нему. В случае инициализации шрифта лучше расположить этот блок до начала основного цикла.

Затем в главном меню следует выбрать пункт `Widgets` и далее – вкладку `Labels`, из которой при помощи мыши перетащить иконку `String` на область дисплея, после чего выбрать в перечне параметров диспетчера объектов для объекта `Strings1` строку позиции `Strings`, как показано на рисунке 10. В столбце `Value`, щёлкнув на кнопке выпадающего меню, открыть редактор шрифтов `Strings Editor` (см. рис. 11). В поле `Font` открывшегося редактора необходимо выбрать из выпадающего списка нужный шрифт. Для выбора доступны все шрифты `Windows`. В поле `Size` задаётся размер шрифта. Выбранный шрифт в дальнейшем будет сохранён в памяти µSD-карты только с заданным размером. После нажатия на кнопку `Paste Code` диспетчера объектов в заданном месте области редактора появится блок кода примерно такого вида, как в листинге 1.

Из этого блока нужно убрать строки с функциями `gfx_MoveTo(...)`, `print(...)`, которые в приложении не потребуются. Цвет шрифта можно задать в диспетчере объектов или же непосредственно

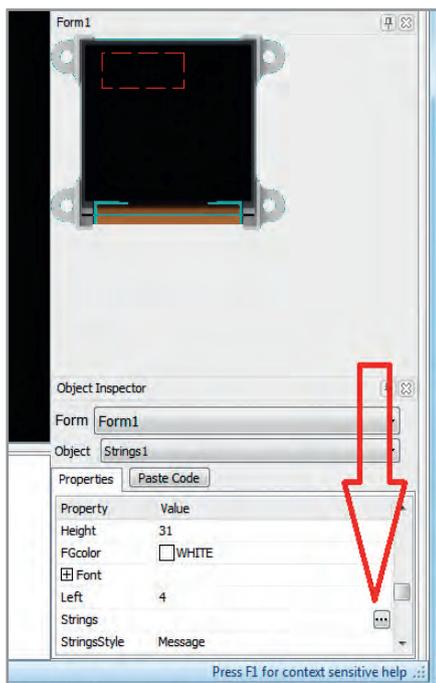


Рис. 10. Создание объекта «текстовый шрифт» Strings1

в исходном тексте, как аргумент функции `txt_FGcolour(...)`. После задания шрифта вывод на дисплей текстовых данных с помощью функций `putnum(...)`, `putstr(...)`, `print(...)` и т.п. будет осуществляться именно этим шрифтом.

Теперь создадим объект «графическая шкала». Блок кода, который будет автоматически вставлен в программу, однократно выводит на дисплей изображение шкалы с количеством засвеченных штрихов, соответствующим значению входной переменной. По этой причине указанный блок следует разместить внутри основного цикла и передавать на него управление с требуемой частотой обновления показаний шкалы.

В окне редактора открытого файла исходного текста выбираем в главном меню пункт `Widgets`, затем – вклад-

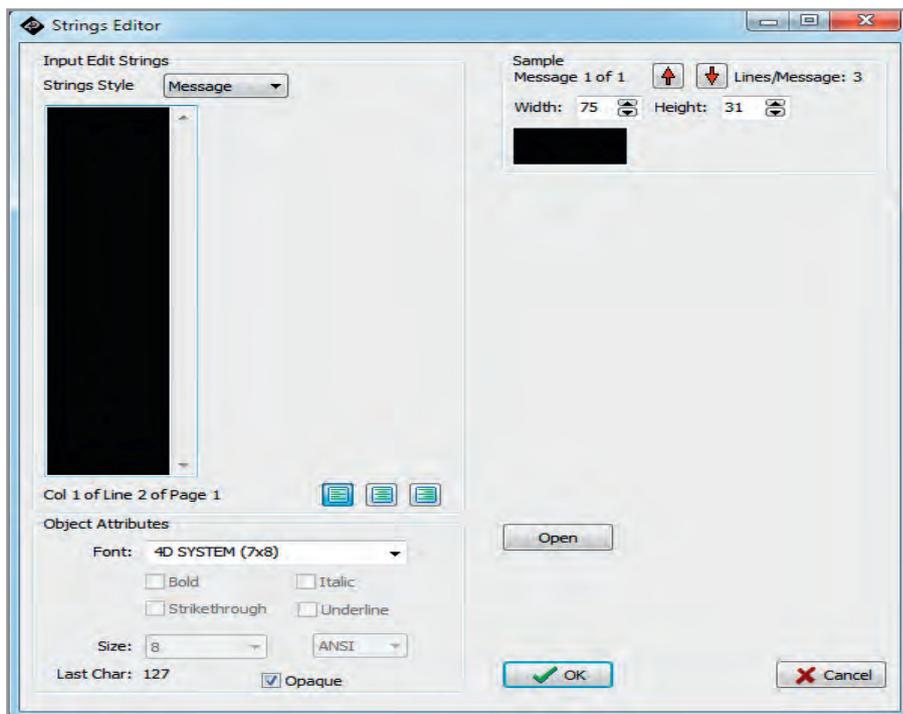


Рис. 11. Окно редактора шрифтов Strings Editor

Листинг 1

```
// Strings1 1.0 generated 25.09.2016 13:23:05
media_SetSector(0, Strings1FontStartL) ; // must come b4 setting
fontID
txt_FontID(MEDIA) ; // Font index correct at time of code generation
txt_FGcolour(RED) ;
txt_BGcolour(BLACK) ;
gfx_MoveTo(4 , 4) ;
print(«Your text goes here») ;
```

ку `Gauges`, из которой перетаскиваем мышью на область дисплея модуля иконку шкалы `Gauge` (см. рис. 12). При помощи мыши можно скорректировать размеры изображения шкалы и расположить его на дисплее нужным образом. Затем в диспетчере объектов можно изменить различные параметры

объекта `Gauge1` (цвет и т.д.). Как можно видеть, по умолчанию шкала имеет три области, каждой из которых задано по два цвета, соответствующие засвеченному и не засвеченному состояниям. Информация о цветах располагается в папках `Palette1–Palette3`. В нашем варианте во всех трёх папках исполь-



НАУЧНО-ПРОИЗВОДСТВЕННОЕ ПРЕДПРИЯТИЕ

РОССИЙСКИЙ РАЗРАБОТЧИК И ПРОИЗВОДИТЕЛЬ

- Разработка герметичных DC/DC-преобразователей для ответственных применений
- Разработка и производство мощных источников питания для авиационной аппаратуры
- Разработка заказных силовых и ВЧ/СВЧ-модулей
- Производство дискретных силовых компонентов в керамических корпусах
- Разработка и проведение испытаний изделий и компонентов силовой электроники



POWERED BY PROSOFT

ОФИЦИАЛЬНЫЙ ДИЛЕР НПП «АРБЕЛОС»

ТЕЛ.: (495) 232-2522 / ФАКС: (495) 234-0640 / INFO@PROCHIP.RU / WWW.PROCHIP.RU

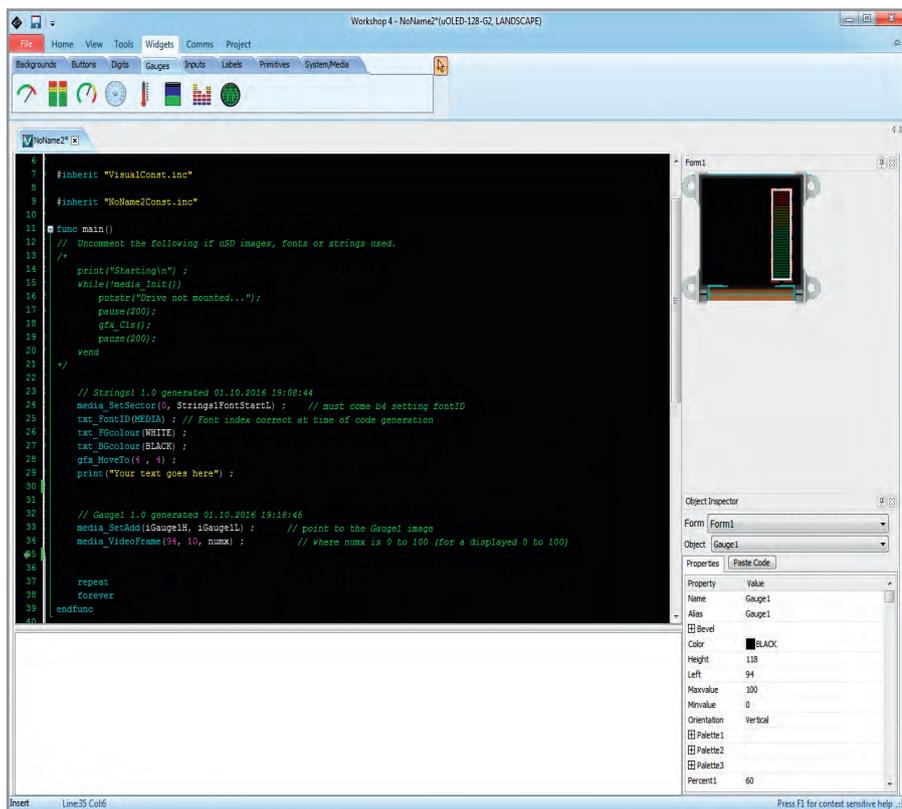


Рис. 12. Создание объекта «графическая шкала» Gauge1

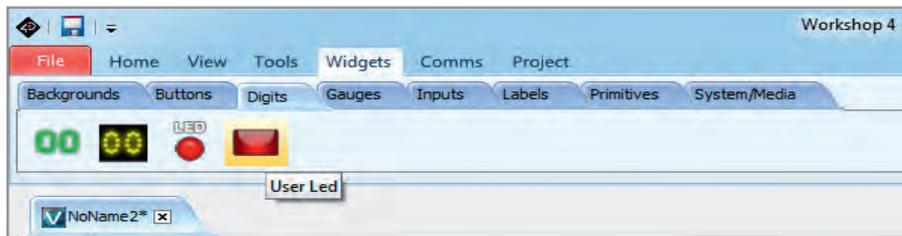


Рис. 13. Создание объекта «светодиод» User Led1

Листинг 2

```
// Gauge1 1.0 generated 01.10.2016 19:18:45
media_SetAdd(iGauge1H, iGauge1L) ; // point to the Gauge1 image
media_VideoFrame(94, 10, numx) ; // where numx is 0 to 100
```

Листинг 3

```
// Userled1 1.0 generated 01.10.2016 19:26:08
media_SetAdd(iUserled1H, iUserled1L) ; // point to the Userled1 image
media_VideoFrame(15, 84, numx) ; // where numx is 0 (Off) or 1 (On)
```

зуется одинаковая пара цветов, поэтому на дисплее шкала будет выглядеть однородной. Здесь же при необходимости можно скорректировать параметры Minvalue и Maxvalue, определяющие диапазон входной переменной шкалы.

По завершении редактирования и нажатии на кнопку Paste Code диспетчера объектов в области редактора появится код, приведённый в листинге 2.

Пользовательское редактирование этого фрагмента состоит в указании входной переменной вместо заданной по умолчанию переменной numx. Значения констант iGauge1H, iGauge1L, определяющих местоположение объекта в памяти носителя, генерируются автоматически в заголовочном файле констант проекта. Созданная по умолчанию в шаблоне ссылка на заголо-

вочный файл выглядит как #inherit "voltmeter_1Const.inc". В этом файле ранее была сгенерирована константа Strings1FontStartL, определяющая местоположение заданного ранее шрифта. Нажав правой кнопкой мыши на ссылку, при необходимости можно командой Open file at Cursor открыть этот файл для просмотра и редактирования.

Для создания объекта «светодиод» в шаблоне исходного кода нужно поместить курсор в соответствующее место в программе. Блок кода, который будет туда автоматически вставлен, выводит на дисплей изображение светодиодного индикатора. В зависимости от значения входной переменной (0 или 1) индикатор будет погашен или зажжён. Управление должно передаваться в этот блок только тогда, когда требуется зажечь или погасить «светодиод».

В окне редактора следует выбрать в главном меню пункт Widgets, затем – вкладку Digits, из которой перетащить мышью на область дисплея модуля иконку прямоугольного светодиода User Led (см. рис. 13). Позже можно изменить размеры и местоположение изображения.

По завершении редактирования нажатием на кнопку Paste Code диспетчера объектов в программу добавится код, показанный в листинге 3.

В добавленном коде необходимо указать входную переменную вместо установленной по умолчанию переменной numx.

Для удаления какого-либо графического объекта с дисплея и из диспетчера объектов, нужно выделить его изображение на дисплее и нажать на клавиатуре клавишу Del, после чего удалить вручную соответствующий этому объекту блок кода.

Часть графических объектов, например, табло семисегментных индикаторов Led Digits, Custom Digits с вкладки Digits, для корректной компиляции требуют наличия в программе ссылки #inherit "LEDDIGITSDISPLAYGOLDELOX.INC". Если в приложении эти объекты используются, а в ходе проектирования указанная ссылка не создалась автоматически, её следует создать вручную.

ЛИТЕРАТУРА

1. www.4dsystems.com.au/product/4D_Workshop_4_IDE.
2. 4DGL Programmers Reference Manual and Language Specifications[©] 2009 4D Labs.
3. 4D Systems Goldelox 4DGL Internal Functions.
4. 4D Systems Goldelox Serial Environment Command Set. Part of the Workshop 4 IDE.

