Комплект разработки

программы Zvonok

Содержание

[1. Аннотация 1](#_Toc112214636)

[2. Состав комплекта 1](#_Toc112214637)

[3. Настройка 1](#_Toc112214638)

[4. Установка на ПК 2](#_Toc112214639)

[5. Описание работы 2](#_Toc112214640)

[6. Отладка 4](#_Toc112214641)

# Аннотация

Программа предназначена для управления подачей звуковых и аудио сигналов (сообщений) в соответствии с задаваемым расписанием.

Программа работает под ОС Microsoft Windows 7, 10.

# Состав комплекта

* 1. distributive – дистрибутив.
* Borland C++ Builder Enterprise 6.0;
* SetupVCD5500;
  1. documents – документация;
* Описание Zvonok.docx;
  1. CBsource – исходный код программы для Borland C++ Builder.

# Настройка

* 1. Разработка производилась под ОС Microsoft Windows 7 Service Pack 1;
  2. Установите на ПК эмулятор CDSetupVCD5500.
  3. Установите BorlandC++ Builder подключая образа CDдисков через эмулятор CD.

- подключите образ диска Borland C++ Builder 6.0 Enterprise [cd1].iso.

- запустите на виртуальном CD программу install.exe.

- в программе установки выберите пакет«C++ Builder 6».

- при запросе программы установки «SerialNumber», запустите программу Borland C++ Builder 6.0 keygenerator.exe,

- в программе Borland Keygen нажмите кнопку generate.

- скопируйте в программу установки сгенерированные коды.

- далее следуйте программе установки,

- по запросу программы установки замените на виртуальном CD образ диска Borland C++ Builder 6.0 Enterprise [cd2].iso.

* 1. При первом запуске C++ Builder выберите вариант регистрации «Register or activate by phone…»
* запустите программу Borland C++ Builder 6.0 keygenerator.exe,
* скопируйте коды из программы Borland C++ Builder в программу Borland Keygen, получите ActivationKey и укажите его в программе C++ Builder.
  1. ВC++ Builder открыть проект Zvonok.bpr из каталога CBsource.

# Установка на ПК

* 1. Скопируйте каталог Release и запустите файл Zvonok.exe

# Описание работы

* 1. Настройка
     1. При запуске программа считывает настройки из файла setup.ini. При его отсутствии - из файла старого формата par.txt. Если нет никаких файлов, то параметры настройки устанавливаются по умолчанию.

Настройки хранятся в структуре set\_setup.

* + 1. При запуске, в методе FormCreate программа устанавливает значение задержки включения усилителя звукового сигнала в поле структуры

set\_setup.up\_booster\_delay = DEF\_UP\_BOOSTER\_DELAY

, значение 3 по умолчанию.

В функции OpenFiles программа считывает значение задержки из файла. Если при открытии файла, его чтении или сохранении диапазона значений (1..99) возникают ошибки, программа оставляет значение по умолчанию.

* + 1. При запуске, в методе FormCreate программа устанавливает значение задержки выключения усилителя звукового сигнала в поле структуры

set\_setup.down\_booster\_delay = DEF\_DOWN\_BOOSTER\_DELAY

, значение 1 по умолчанию.

В функции OpenFiles программа считывает значение задержки из файла. Если при открытии файла, его чтении или сохранении диапазона значений (1..99) возникают ошибки, программа оставляет значение по умолчанию.

* + 1. При запуске, в методе FormCreate программа устанавливает значение часового пояса в поле структуры

set\_setup.time\_zone = DEF\_TIME\_ZONE

, значение 3 по умолчанию.

В функции OpenFiles программа считывает значение часового пояса из файла. Если при открытии файла, его чтении или сохранении диапазона значений (2.12) возникают ошибки, программа оставляет значение по умолчанию.

* + 1. При запуске, в методе FormCreate программа устанавливает значение флага перехода на зимнее время в поле структуры

set\_setup.winter\_time = DEF\_WINTER\_TIME

, значение 0 по умолчанию.

В функции OpenFiles программа считывает значение флага из файла. Если при открытии файла, его чтении или сохранении диапазона значений (0..1) возникают ошибки, программа оставляет значение по умолчанию.

* + 1. При запуске, в методе FormCreate программа устанавливает имя звукового файла для внешнего сигнала «Тревога» в поле структуры

set\_setup.alarm\_audio\_file = DEF\_ALARM\_SOUND

, значение “Terror.wav” по умолчанию.

Программа проверяет наличие этого файла в директории программы и устанавливает флаг его наличия

set\_setup.are\_sound\_files += IS\_DEF\_ALARM

В функции OpenFiles программа считывает имя звукового файла “Тревога” из файла настроек. Если при открытии файла настрок, его чтении или поиске звукового файла возникают ошибки, программа оставляет значение по умолчанию.

* + 1. При запуске, в методе FormCreate программа устанавливает имя звукового файла для внешнего сигнала «Пожар» в поле структуры

set\_setup.fire\_audio\_file = DEF\_FIRE\_SOUND

, значение “Fire.wav” по умолчанию.

Программа проверяет наличие этого файла в директории программы и устанавливает флаг его наличия

set\_setup.are\_sound\_files += IS\_DEF\_FIRE

В функции OpenFiles программа считывает имя звукового файла “Пожар” из файла настроек. Если при открытии файла настроек, его чтении или поиске звукового файла возникают ошибки, программа оставляет значение по умолчанию.

* + 1. При запуске, в методе FormCreate программа устанавливает имя звукового файла для тестового звукового сигнала в поле структуры

set\_setup.test\_audio\_file = DEF\_TEST\_SOUND

, значение “Ring.wav” по умолчанию.

Программа проверяет наличие этого файла в директории программы и устанавливает флаг его наличия

set\_setup.are\_sound\_files += IS\_DEF\_TEST\_SOUND

В функции OpenFiles программа считывает имя тестового звукового файла из файла настроек. Если при открытии файла настроек, его чтении или поиске звукового файла возникают ошибки, программа оставляет значение по умолчанию.

* + 1. Если при запуске программа не обнаружила звуковых файлов, указанных в файле настроек или не обнаружила звуковых файлов заданных по умолчанию, то программа отображает окно с выводом ошибок.
    2. Программа считывает расписание событий из файла Schedule.sch.

Формат файла:

#

<параметры события>

<имя файла звукового сообщения>

<строка 1 описания события>

…

<строка n описания события>

#

, где символ # - определяет начало новой записи.

Параметры события записываются в виде строки:

<year> <month> <day> <weekday> <use\_year> <use\_month> <use\_day> <use\_weekly> <ex\_day>

<use\_hour> <use\_minute> <use\_second> <hour> <minute> <second>

<cycle\_flags> <event\_sign> <row\_count>

, где параметры разделены пробелом, а количество символов каждого параметра произвольное,

параметры описаны в структуре t\_Event,

row\_count – количество строк описания события (n)

При отсутствии файла расписания программа может считать расписание из файлов старого формата: day.txt, rxx.txt (где хх – от 00 до 99), ex.txt

* + 1. Если при запуске программа не обнаружила файлов расписания, в окне программы будет отображаться сообщение «Добавьте событие…»
  1. Формирование расписания
     1. Все события считанные из файла сохраняются в векторе BaseEvent.

* + 1. Программа не должна обрабатывать пропущенные события при перезапуске.

# Замечания

1. Функция malloc работает не стабильно. При ее использовании при закрытии программы появляется ошибка при попытке освободить память – лучше не использовать.
2. Для элемента CheckBox при ручном изменении параметра Checked автоматически вызывается функция Click().
3. нет обработки многострочного соообщения
4. нет возможности пролистывать расписания по дням вперед-назад
5. при добавлении события нет списка событий на планируемый день (конфликт событий)
6. события с пустыми параметрами день время не срабатывают
7. при добавлении события учитывать какой день просматривается
8. для событий без указания дня параметры даты сохраняются в файл
9. для режима удаления – контроль номера события в векторе
10. сделать возврат к текущему списку кликом под отображаемой дате
11. нет контроля перехода суток - смена текущего списка
12. при считывании базы устанавливать признак наличия сегодня праздника
13. длинные строки сообщения при считывании из файла урезаются