УДК 004.357

ПРОГРАММНО-АППАРАТНАЯ СИСТЕМА ПОДАЧИ ШКОЛЬНЫХ ЗВОНКОВ

Шевченко А.С.
Донецкий лицей «Интеллект»

В наше время все процессы, требующие управления, стараются автоматизировать, будь то работа стиральной машины, холодильника или кондиционера. При этом вмешательство человека стараются свести к минимуму.

Исходя из этого родилась идея автоматизировать подачу школьных звонков. Устройством, управляющим подачей звонков, выбран стационарный компьютер, поскольку это позволит при необходимости расширить набор функций программы, подающей звонки.

С компьютера через COM-порт сигнал поступает на рабочую плату, где замыкается электрическая цепь, содержащая школьный звонок.

В основе программы лежит асинхронный способ работы COM-порта. Операционная система Windows позволяет одновременно запускать несколько программ и, кажется, что все они работают одновременно. В действительности в каждый момент времени (на однопроцессорном компьютере) выполняется только одна программа. Windows выделяет для каждой программы маленький участок времени, называемый квантом, в течение которого программа работает. После того, как участок квант времени закончится, Windows останавливает ту программу, которая выполнялась и запускает следующую. Так как квант времени мал, то у пользователя создается ощущение, что все программы выполняются одновременно (на двух и более процессорных системах действительно может выполняться 2 и более программы одновременно). На самом деле кванты времени выделяются не программам, а потокам. Любая программа обязательно содержит хотя бы один поток, но может содержать и больше одного потока. Если создать в программе больше одного потока, можно добиться того, что пользователю будет казаться будто бы программа выполняет несколько задач одновременно (а на многопроцессорных системах задачи могут выполняться действительно одновременно).

Из вышесказанного следует, что в обычном режиме работы пользователь сам должен выбирать какой поток будет создан: считывающий или передающий. Можно было бы постоянно опрашивать COM-порт в бесконечном цикле, но тогда наша программа скорей всего не будет реагировать на действия пользователя и будет сильно загружать процессор. Для того, чтобы пользователь мог управлять программой, опрос порта нужно вынести в отдельный поток, однако это не решит проблемы с загрузкой процессора. Тут нам на помощь приходят события. При опросе COM-порта функция чтения выносится в отдельный поток. Поток чтения нам не нужен, мы работаем с потоком записи. Но это не все потоки, которые нам нужны.

Один поток следит за датой и берёт её значение, другой — каждые несколько часов берёт расписание звонков из файла-расписания и заполняет ими переменные, третий — сравнивает заданное расписание с текущим временем и в случаи совпадения передаёт управления четвёртому потоку, который, в свою очередь, отправляет заранее определённый в программном коде массив данных (в нашем случаи это массив одинаковых символов с кодом «255», что в битовом представлении 11111111). Этот массив единиц обеспечит нам на выходе COM-порта нужное напряжение.

Сама программа написана на языке программирования C++. При запуске программы сразу работает несколько параллельных и независимых потоков. Первый опрашивает компьютер на наличие хотя бы одного COM-порта. В случае обнаружения интересующего порта программа продолжает работу потока, который должен задать все константы, параметры и функции COM-порта, которые нужны для работы системы, иначе выдать ошибку и приостановит выполнение данного потока. Для начала нужно инициализировать непосредственно тот порт, который будет выполнять функцию управления, далее идет создание и инициализация DCB-структуры. (Структура DCB определяет установку управления для последовательного порта ввода-вывода). Эта структура содержит все настройки COM-порта, а именно: скорость передачи данных (в нашем случае 10 бод), включается

двоичный режим обмена и т.д.

Дальше поток должен установить блок параметров COM-порта. Эти параметры носят название таймаут.

Нас интересуют не все параметры, поэтому установим некоторые из них: таймаут между двумя символами, общий таймаут операции записи и константа для общего таймаута операции записи. Есть ещё несколько параметров таймаута, но они важны только в режиме чтения. На все таймауты, которые мы задали, ставим значение 0. После этого вся структура таймаутов заносится в COM-порт, который после всего этого ещё раз проверяется на работоспособность. После проделанных операций наш COM-порт готов к работе.

Другой поток принимается за считывания расписания звонков из файла.

Сначала планировалось, что программа будет работать с одним расписанием звонков, но в некоторых школьных заведениях существует несколько расписаний звонков. То есть на некоторые дни недели существует расписание звонков, которое несколько отличается от стандартного.

И исходя из этого, поток берёт из компьютера день недели и открывает соответствующий файл с расписанием звонков. Считанными значениями он заполняет переменные, которые были специально созданы для этих целей. И на этом действие потока прекращается.

После создаётся сразу ещё несколько потоков.

Первый поток раз в минуту (интервал возможно изменять в коде программы) сравнивает текущее время с массивом расписания, и в случаи совпадения идет программное нажатие кнопки «Подать звонок». При нажатии данной кнопки происходит открытие COM-порта и очистка передающего буфера, после чего происходит передача массива данных в виде трёх символов, код которого «255». Если передача прошла успешно, то по окончании передачи данных подаётся команда закрыть COM-порт.

Второй поток один раз в секунду берёт значение времени с компьютера и выводит его на экран.

Все выполняемые операции отображаются на форме (Form1) в компоненте Memo1 (рис. 1).

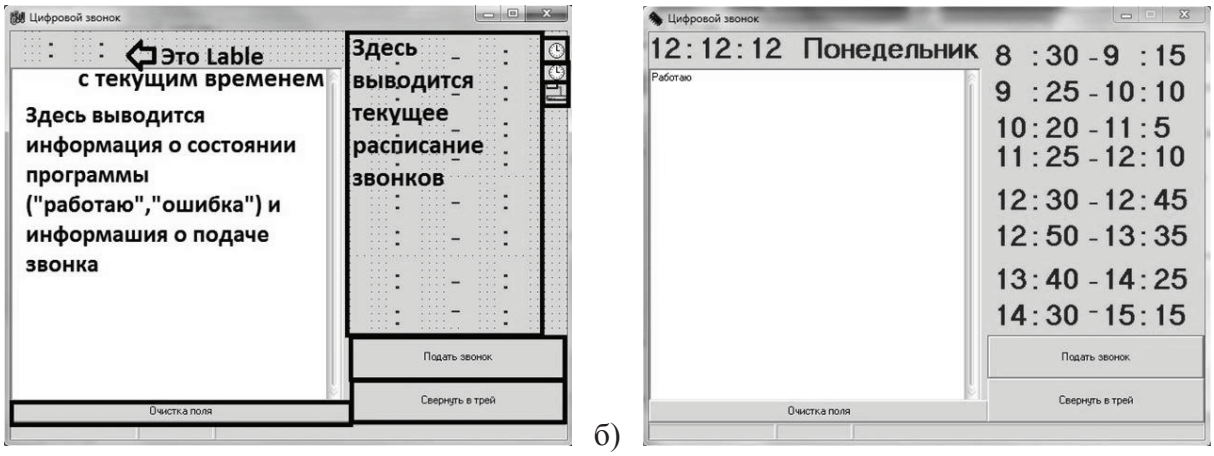

Рисунок 1. Внешний вид формы(Form1) программы:а) схематическое расположение объектов на форме; б) вид формы при работающей программе.

Существует кнопка очистки (с именем «Очистка поля»), которая должна очищать весь блок Memo1.

Программа умеет работать в фоновом режиме, для включения которого нужно нажать кнопку «Свернуть в трей», после чего форма программы закрывается, появляется иконка в трее, но программа продолжает работать в нормальном режиме.

Схемная реализация.

Рассмотрим работу прилагаемой схемы (рис. 2).

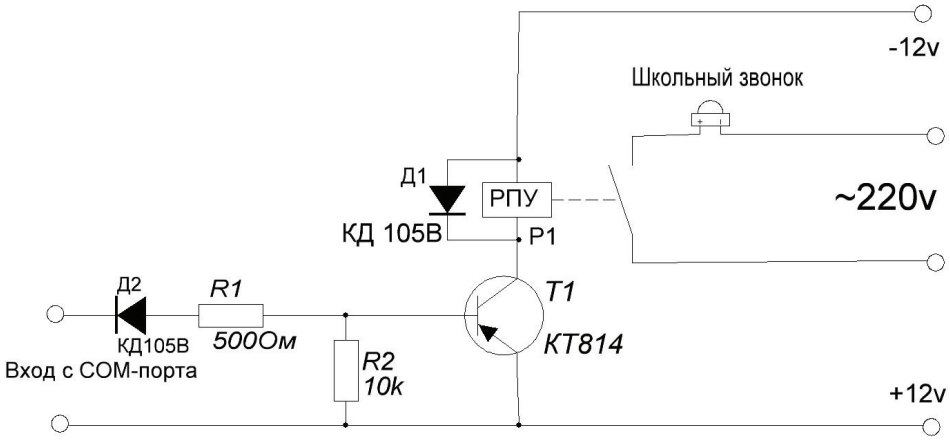

Рисунок 2. Схема устройства.

С COM-порта, а точнее с выводов 3 и 5, сигнал подаётся на вход электронного блока.

Транзистор T1 работает в режиме электронного ключа. Резисторы R1 и R2 обеспечивают нужный режим работы транзистора. Диод Д2 служит защитой от обратного напряжения подводимого с COM-порта.

В момент поступления сигнала на резистор вход устройства, в цепочку Д1 R1 и соответственно на базу транзистора T1, последний открывается и на реле P1 подаётся напряжение.

Реле срабатывает и с помощью своих НО контактов замыкает кнопку школьного звонка.

Конструктивно у реле имеется три переключающих контакта. Для надёжности срабатывания контакты включены параллельно, что обеспечивает более надёжную работу схемы.

Для электропитания схемы использован стандартный трансформатор 220/12. К вторичной обмотке трансформатора подключен диодный мост. Для сглаживания пульсаций электрического тока использован электролитический конденсатор 200 мкФ х 25В

Внешний источник питания на схеме не показан, т.к. он может быть любым, подходящим по характеристикам.

Литература

[1] Работа с COM портом в Delphi при помощи WinApi. Часть 2 – Асинхронный режим. http://www.compconnect.ru/2011/02/rabota-s-com-portom-v-delphi-pri-pomoshhi-winapi-chast-2-asinxronnyj-rezhim/

[2] Руководство по Последовательному программированию для Операционные системы POSIX/ 5-ый Выпуск, 6-ая Версия Copyright 1994-2005 Майкл Р. Свит

[3] Работа с COM-портом с помощью потоков: статья //PIClist RUS ─ 2007. http://piclist.ru