Москва - 2020

Любопытство. Proteus и Flowchart

Proteus и Flowchart

В.Н. Гололобов

Оглавление

Пролог....2
Первый запуск после установки....3
Визуальное программирование....4
Примеры из новой версии....7
Визуальное программирование – это только «помигать»?....14
Что ещё привлекает внимание при беглом взгляде?....17
Эпилог....20

Пролог

С программой я познакомился, скорее, случайно. Встретив однажды утверждение, что программа хорошо работает только в моделировании цифровых элементов, я решил присмотреться к ней, благо был доступ к «живой», а не «прогулочной» версии. Утверждение, как мне показалось, было лишено оснований, поскольку программа была достаточно широкого действия с целью обучения сквозному проектированию, да и не только, её вполне можно было использовать на практике.

Моё внимание тогда привлекла возможность моделировать устройства на основе микроконтроллеров с внешними компонентами, как цифровыми, так и аналоговыми. Позже, пока была возможность использовать программу, я не однократно обращался к ней в своих сочинениях для радиолюбителей. Задумав однажды написать о программе в качестве героя целой истории, я обнаружил, что программу весьма подробно описывают на одном из сайтов. Что послужило к отказу от моих планов.

Вместе с тем, некоторые заметки о программе я решил не оставлять. Сегодня, когда есть доступ только к пробной версии, ничего более небольшой заметки, думаю, не получится. Но и заметка, возможно, кому-то будет интересна.

Первый запуск после установки

Установка программы «тянет» за собой дополнительные установки. Если не спорить с ассистентом, то вид «приглашения» к работе такой, рис. 1.1.

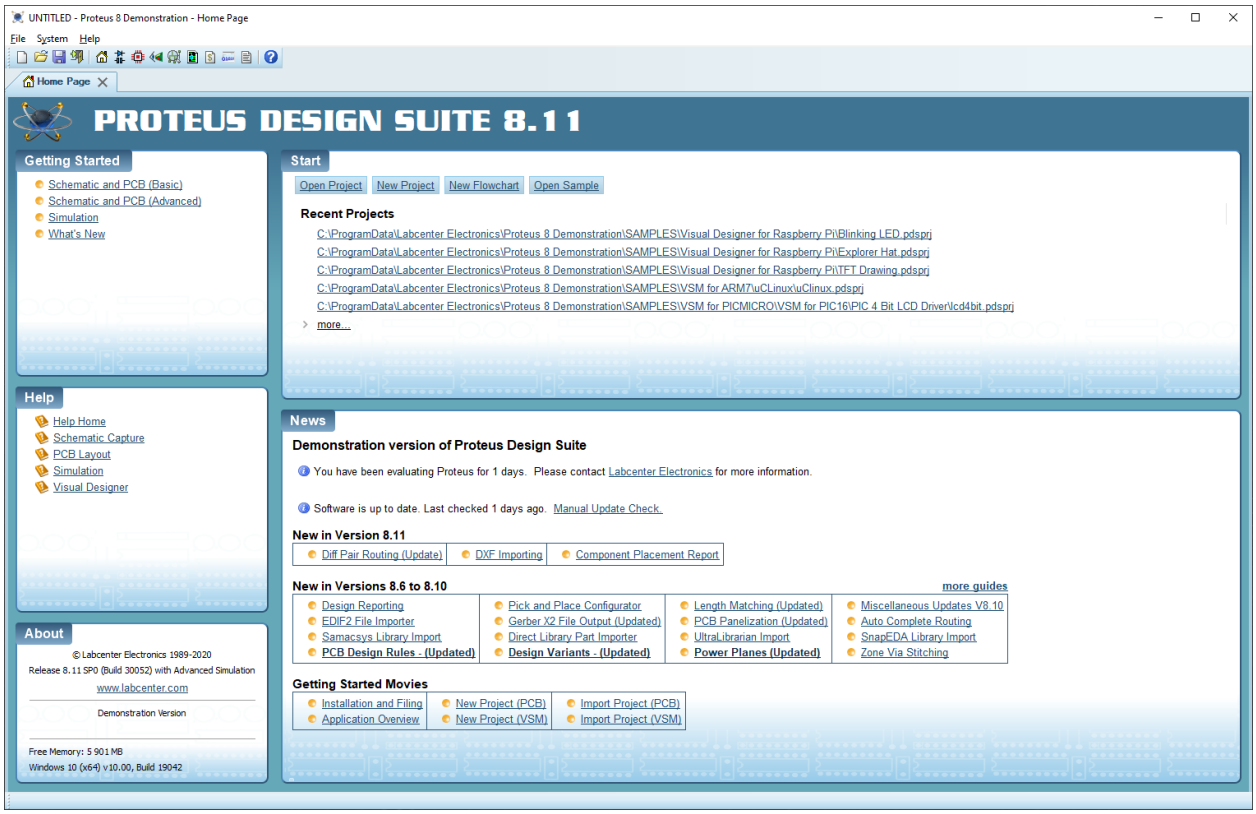

Рис. 1.1. Приглашение к работе Proteus Design Suite 8.11

Если вы знакомы с восьмой версией программы, то новый комплект отличается в первую очередь наличием предложения начать новый Flowchart проект. Что и подвигло «моё любопытство» к установке пробной версии программы.

Есть фирменное описание всех новых функций и особенностей, если вам нужен хороший список модификаций восьмой версии, достаточно воспользоваться ссылкой What’s New в верхней части левой панели.

Особенное внимание советую уделить кнопке «Open Sample». Но меня сейчас более всего интересует визуальное программирование. С него я и хочу начать.

Визуальное программирование

Есть несколько программ, с которыми я знаком хотя бы «шапочно», где используется графический язык программирования. Как бы ни ругали этот вариант программирования микроконтроллеров, он весьма ускоряет работу при создании реальных устройств, а начинающим даёт представление о построении программы (не написании кода). Запускаем новый файл для визуального программирования, рис. 2.1.

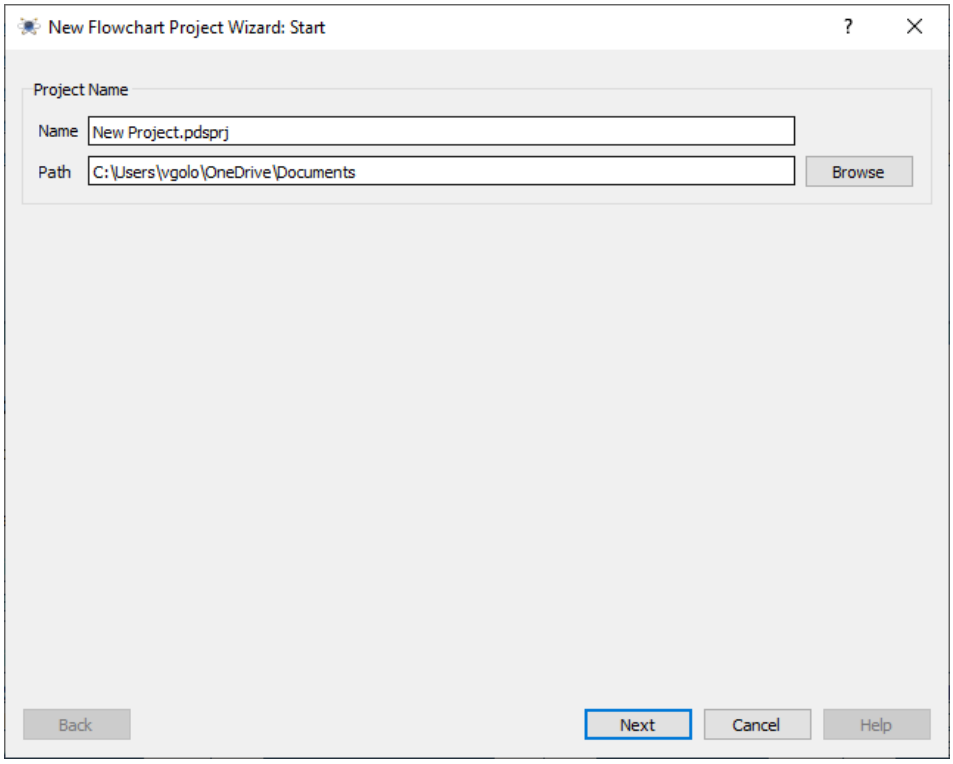

Рис. 2.1. Запуск создания файла кнопкой «New Flowchart»

Помощник создания проекта предлагает дать имя проекту и указать место его расположения, что для меня не имеет смысла, поскольку сохранить проект я не могу, поэтому оставляю всё «как есть». Кнопка «Next» привычное продолжение.

Я могу ошибаться, но визуальное программирование, похоже относится к двум модулям, рис. 2.2.

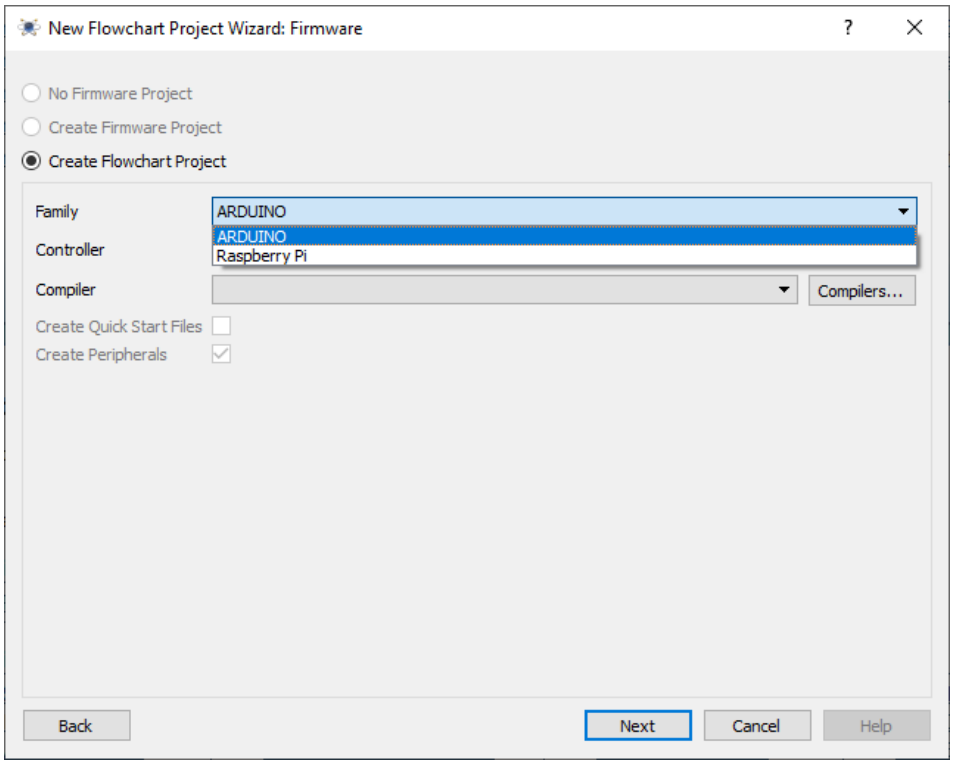

Рис. 2.2. Выбор модуля для нового проекта

Я останавливаю выбор на Arduino. Выбор в окне Controller для меня закрыт, но даёт представление о тех модулях, что доступны в данной версии, рис. 2.3.

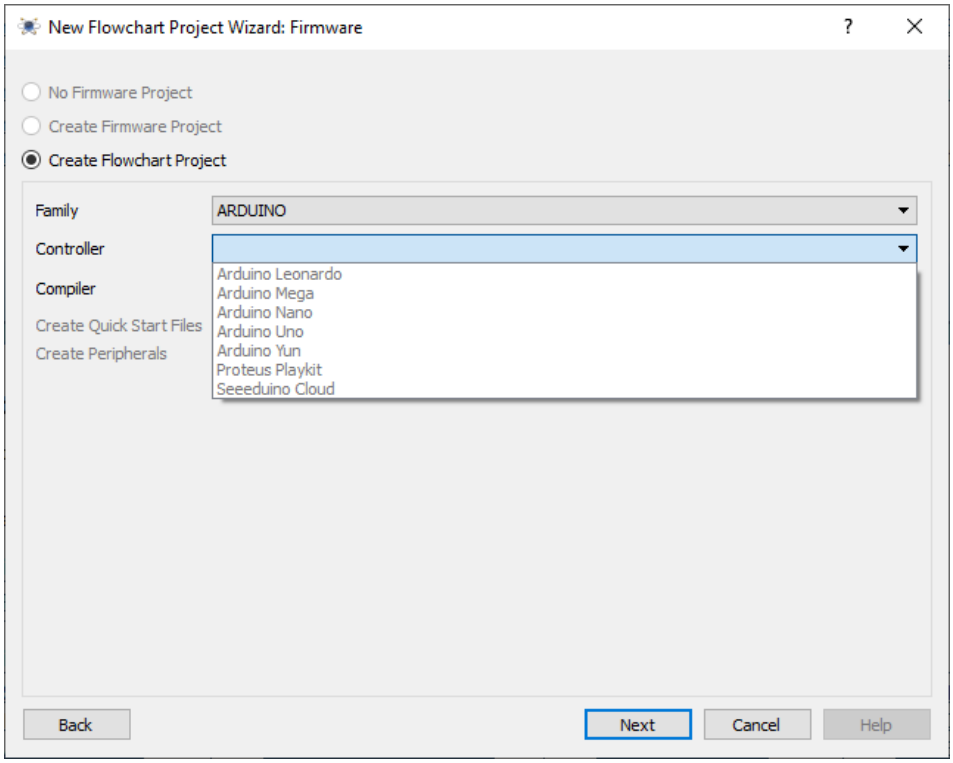

Рис. 2.3. Доступные модули проекта Arduino

Есть ещё окно Compiler. Кнопка для выбора компилятора даёт представление о тех компиляторах, которые вы можете использовать в своих проектах, рис. 2.4.

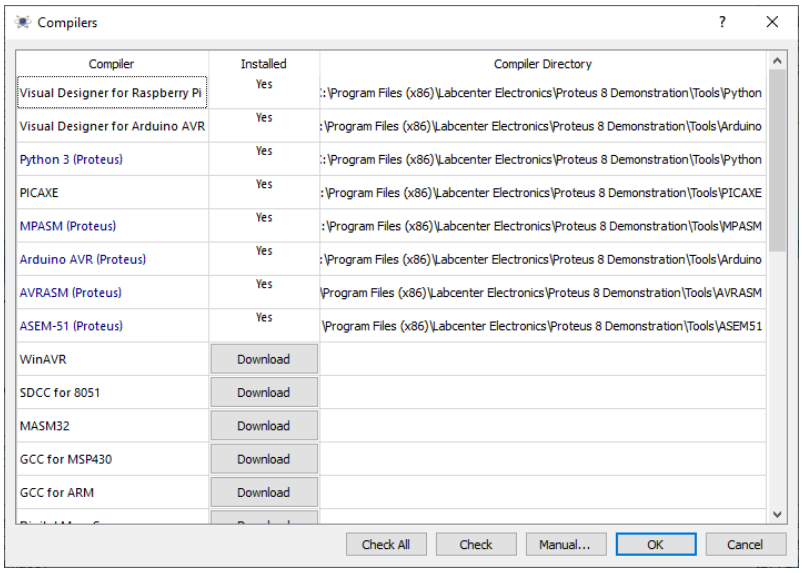

Рис. 2.4. Выбор компилятора для работы с проектом

Я не удержался и выбрал Arduino AVR (чуть ниже в списке), нажал кнопку «Download» и загрузил последнюю (так я думаю) версию программы Arduino. Не думаю, что это было нужно, но попробовать стоило. Следующее окно помощника завершает подготовку. Вы получаете две рабочие области – одна для создания программы, другая для создания схемы. Рабочие области пустые, но, увы, нет средств создания программы для Arduino.

Чтобы хотя бы что-то сказать об этом, я хочу воспользоваться примером. Закрываю первый (неудачный) проект, открываю раздел примеров.

Примеры из новой версии

Не думаю, что у меня хватит терпения просмотреть все примеры, рис. 3.1, но я попытаюсь отыскать пример визуального программирования.

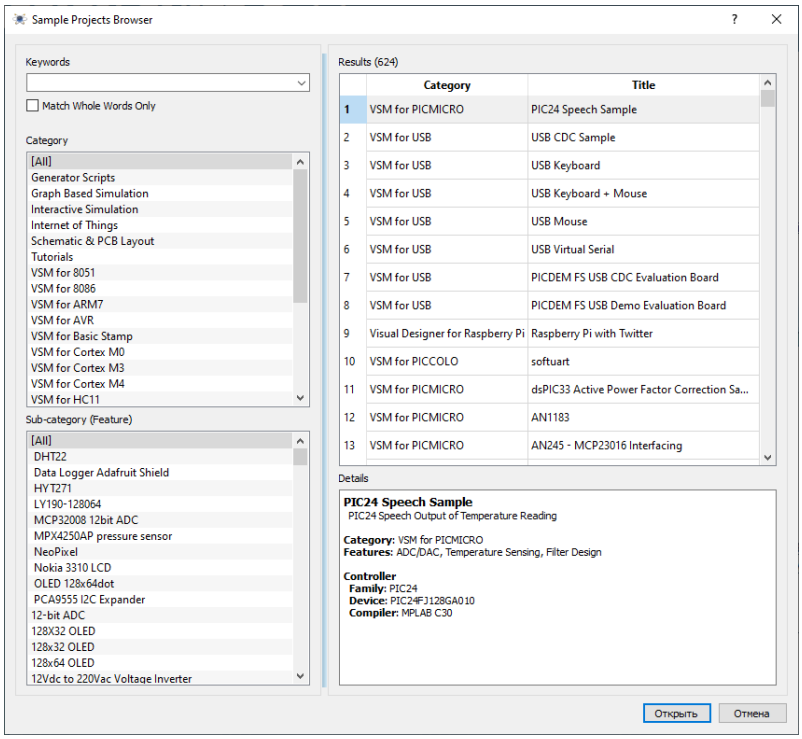

Рис. 3.1. Список доступных примеров

Среди примеров, чтобы не оригинальничать, я нахожу программу «Помигать светодиодом», что даёт представление о происходящем. Появляется схема, которую можно запустить для проверки и отладки, рис. 3.2.

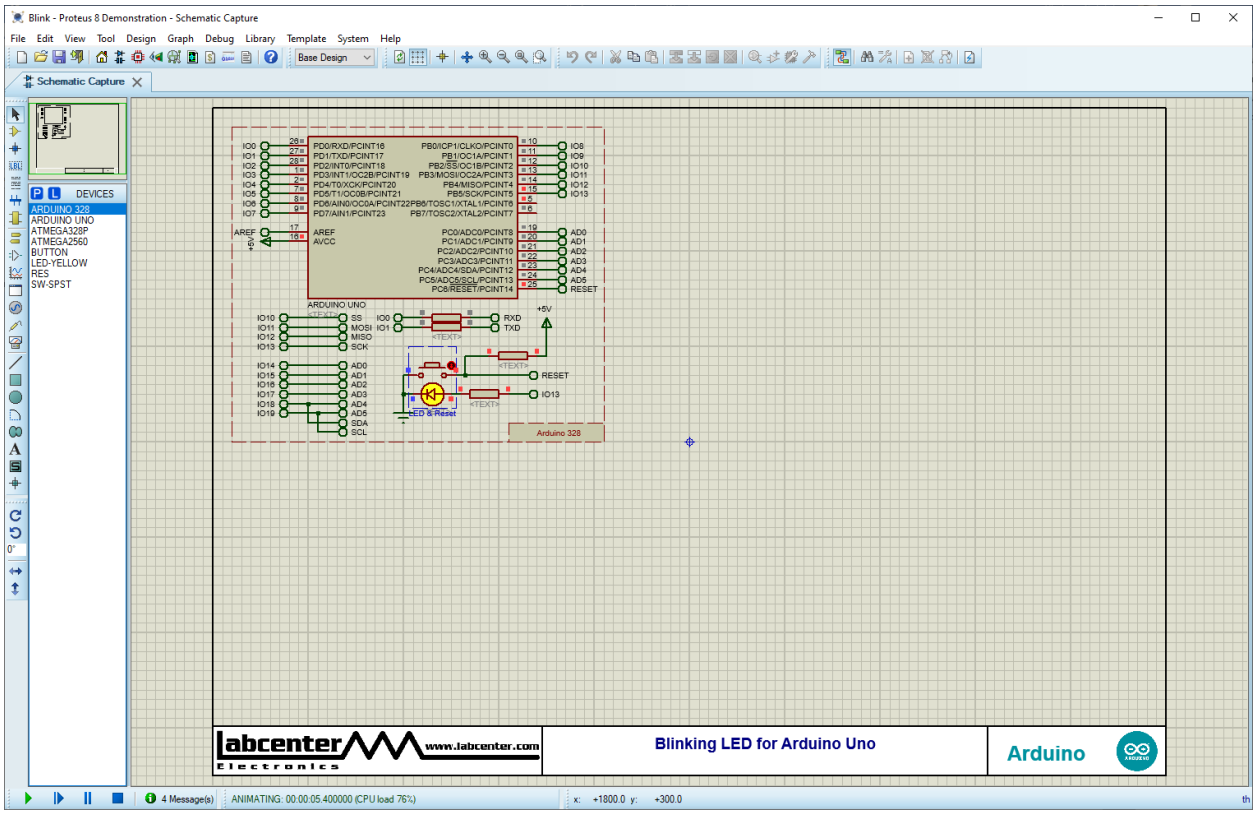

Рис. 3.2. Схема для Arduino со встроенным в модуль светодиодом на выводе 13

Для доступа к программе я использую кнопку в основном инструментальном ряду, под которой появляется подсказка Source Code. Это открывает рабочее поле программирования. Дальше я, что делаю часто, слукавлю: я удалю всё, что есть на странице, чтобы показать, как оно было сделано, рис. 3.3.

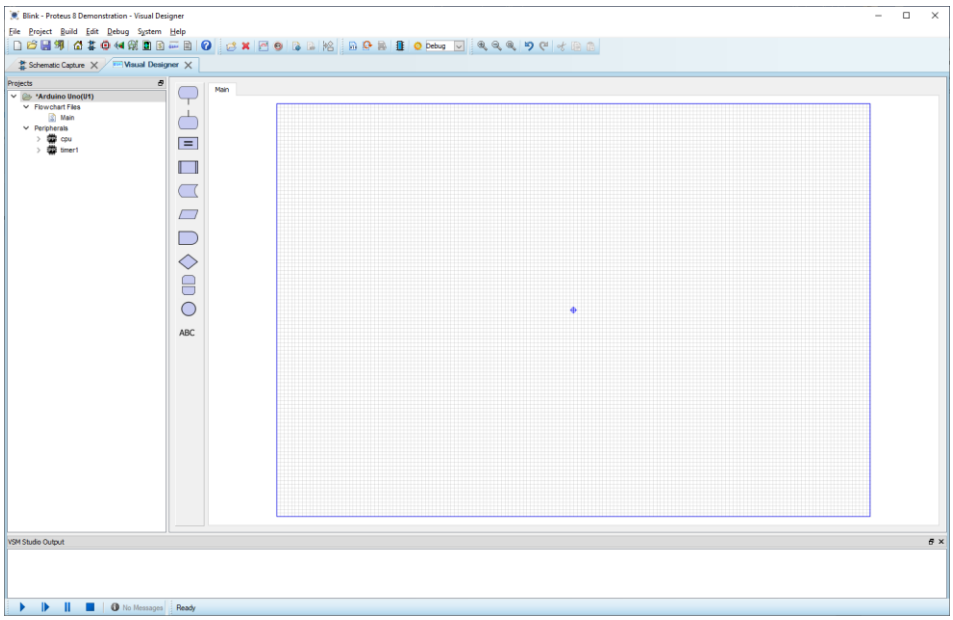

Рис. 3.3. Пустое рабочее поле и инструментальная панель визуального программирования

Я уже подсмотрел, как всё устроено, но сделаю вид, что прочитал подсказку по визуальному программированию. А сделано всё, похоже, соответственно программе Arduino, то есть: выделяется блок начальных установок, к которому добавляется бесконечный цикл.

Итак. Есть элементы начала и конца программного блока. Добавим первый в рабочую область, рис. 3.4.

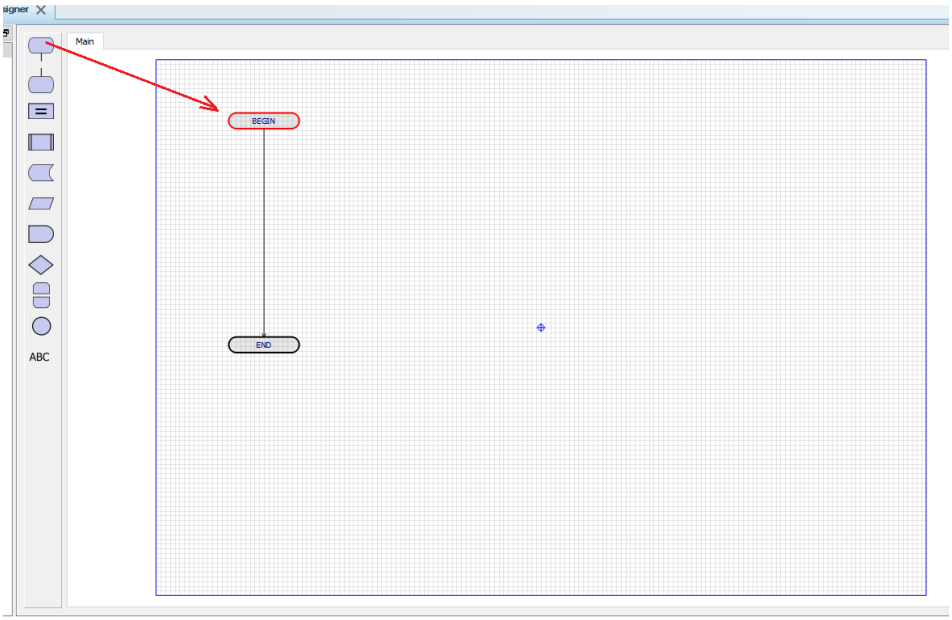

Рис. 3.4. Начинаем создание программы

Щелчком левой клавиши мышки по первому элементу программы можно открыть его свойства, чтобы переименовать его, рис. 3.5.

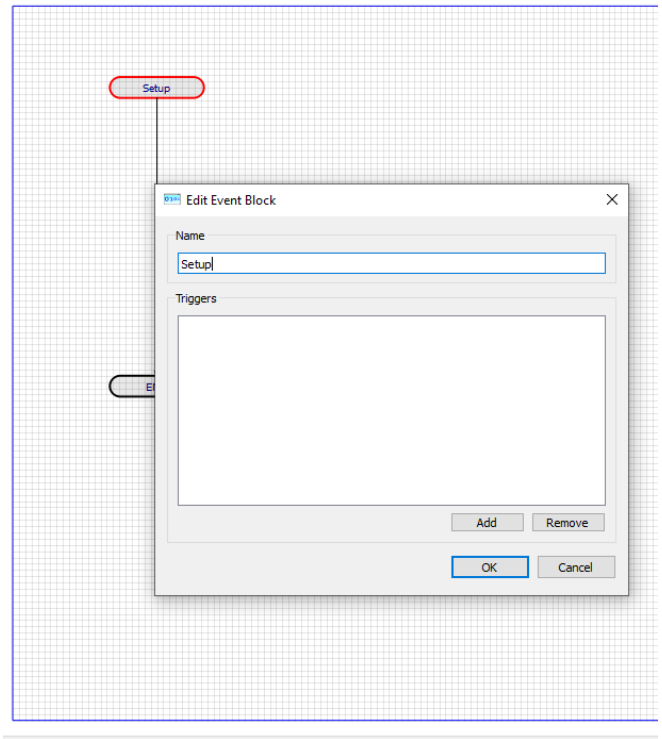

Рис. 3.5. Переименование блока установок

Не берусь утверждать, что это обязательно, но проверять это лучше при использовании полнофункциональной версии. В этом разделе следует направить вывод 13 на выход, на вывод данных, используя блок I/O, рис. 3.6.

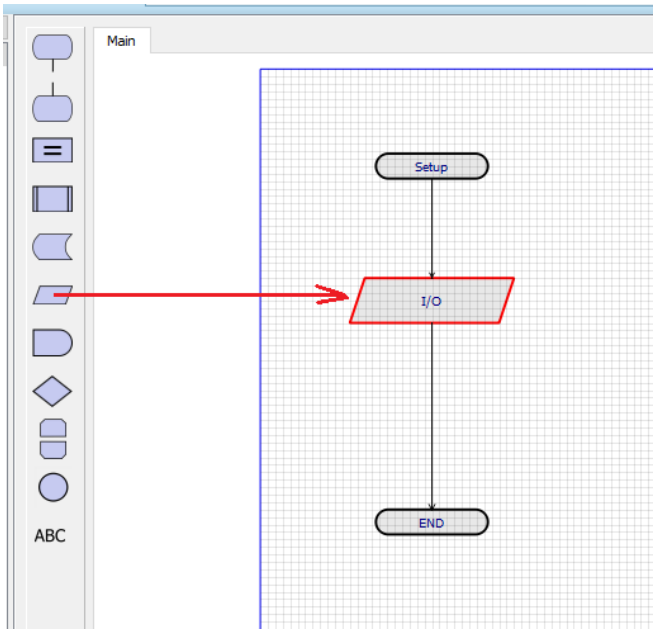

Рис. 3.6. Использование программного блока ввода-вывода

После «перетаскивания блока» в рабочую область следует совместить его точки соединения с линией между ранее установленных блоков. Щелчком по блоку открываем его свойства, где прописываем необходимое, рис. 3.7.

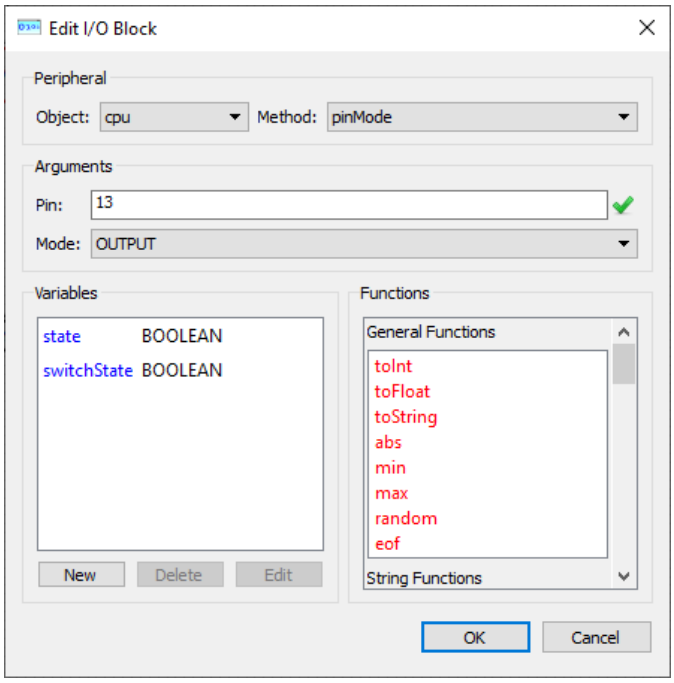

Рис. 3.7. Настройка вывода 13 на выход

На этом заканчивается блок настроек программы. Следующей частью программы будет цикл (loop). Его переносим в рабочую область так же, как ранее «программные скобки» (BEGIN-END), рис. 3.8.

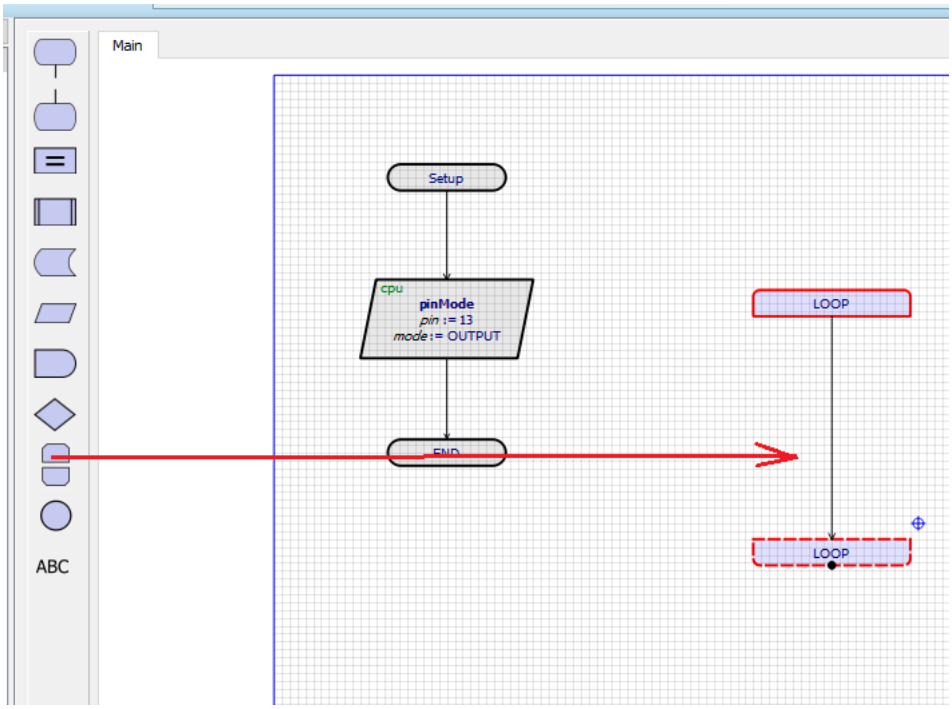

Рис. 3.8. Создание цикла

Но цикл нужно заполнить содержимым. Перенесём блок ввода-вывода, где запишем нужные данные, выбрав нужную функцию, рис. 3.9.

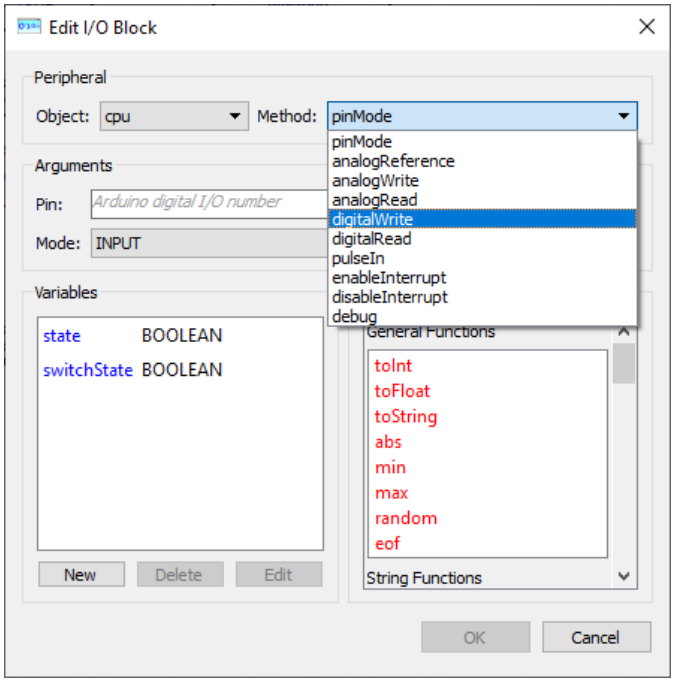

Рис. 3.9. Выбор функции, переводящей вывод 13 в высокое состояние

После выбора функции записи меняется диалоговое окно, где можно ввести нужный вывод, но и нужное состояние вывода, рис. 3.10.

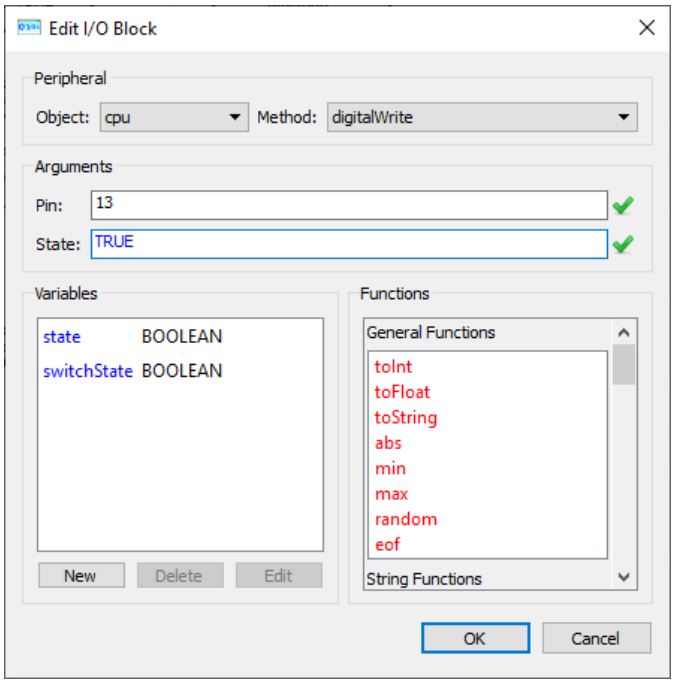

Рис. 3.10. Задание вывода, его состояния и нужной операции

Следом добавим паузу, в свойствах изменим длительность паузы; затем добавим изменение состояния вывода 13 и ещё одну паузу в 1000 мс. Результирующая программа имеет вид, рис. 3.11.

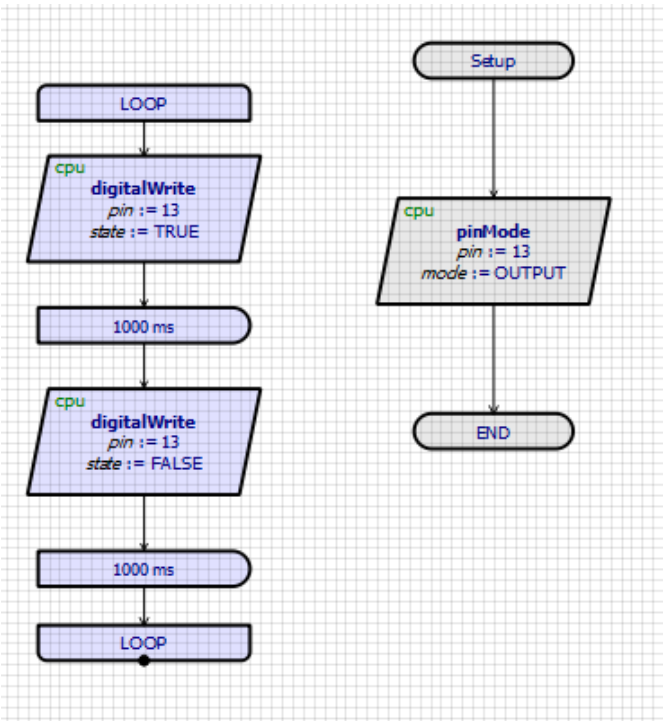

Рис. 3.11. Вся программа «Hello world!»

Построив программу, её можно собрать (Build->Build Project). Об успешной компиляции и сборке программы вы увидите сообщение в окне наблюдения. Программа на странице моделирования схемы не хочет работать, что может быть связано и с моими ошибками, и с неполным функционированием программы, что я проверить, увы, не могу.

Визуальное программирование – это только «помигать»?

Среди сотен примеров есть примеры работы с модулем (компьютером) Raspberry Pi. Вот один из них, рис. 4.1.

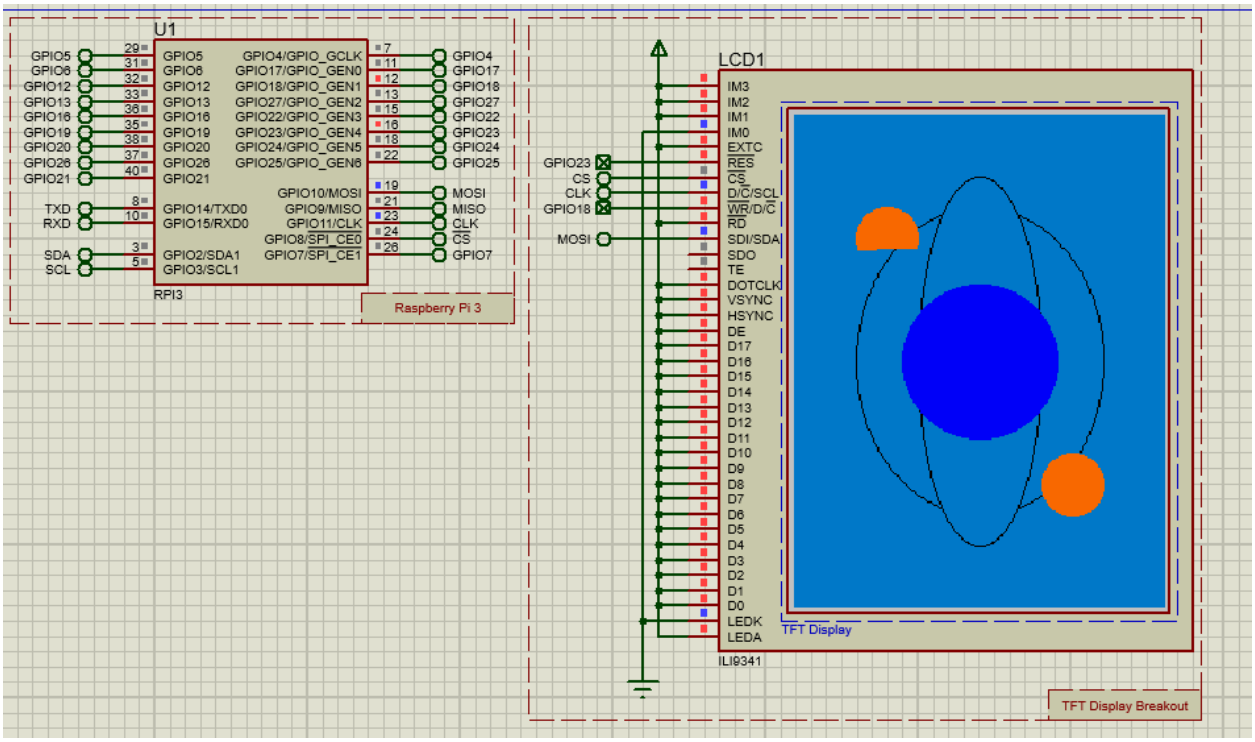

Рис. 4.1. Пример работы с графическим дисплеем

И эта программа создана с помощью визуального программирования, рис. 4.2.

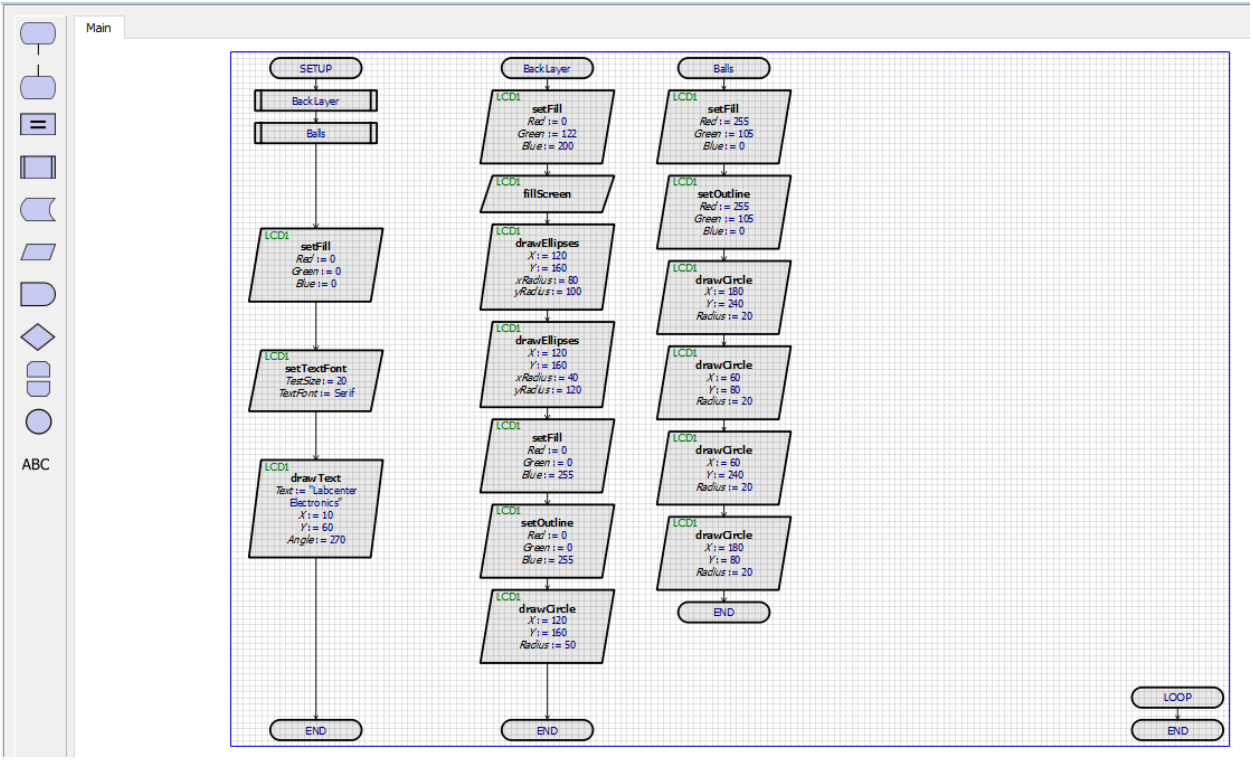

Рис. 4.2. Пример программы для Raspberry Pi

Конечно, если вы намерены познакомиться с программированием на ассемблере, Proteus предоставит такую возможность, рис. 4.3.

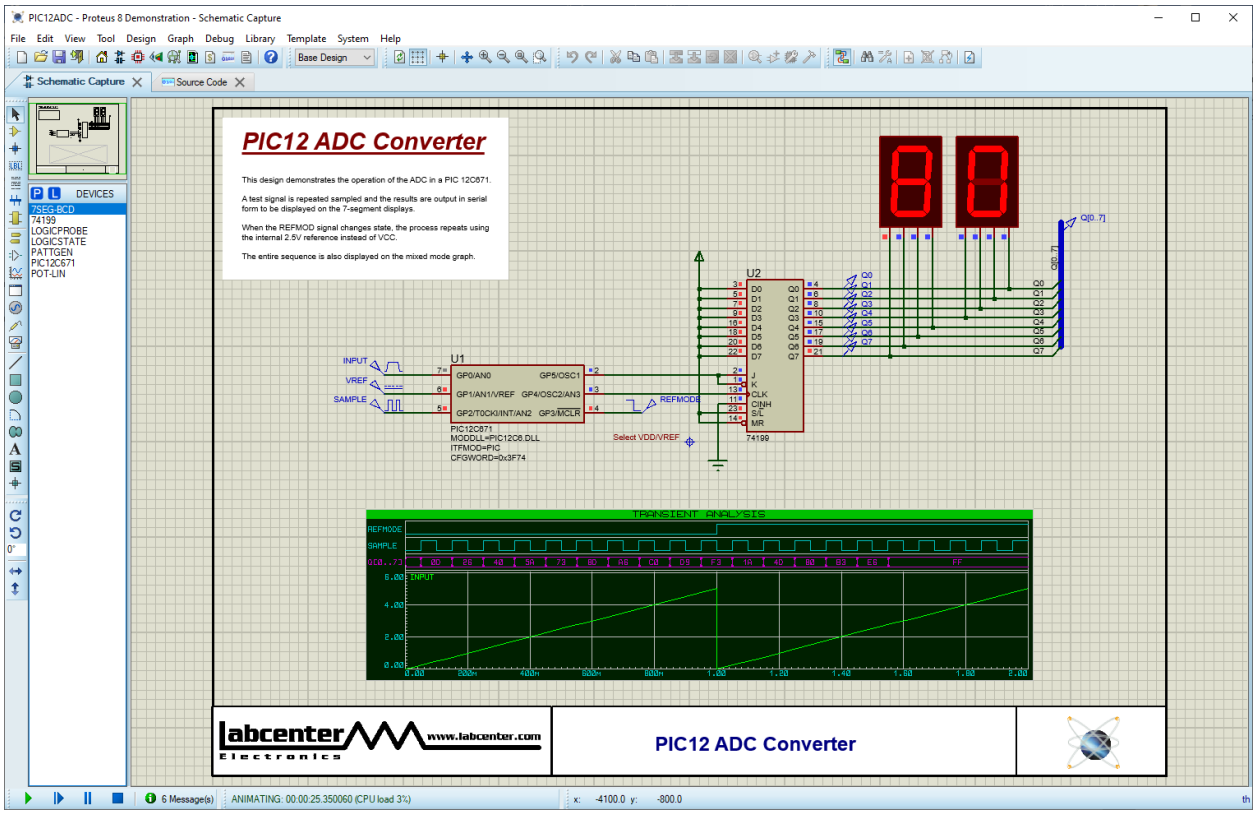

Рис. 4.3. Схема конвертера

Программа в данном случае написана на ассемблере, рис. 4.4.

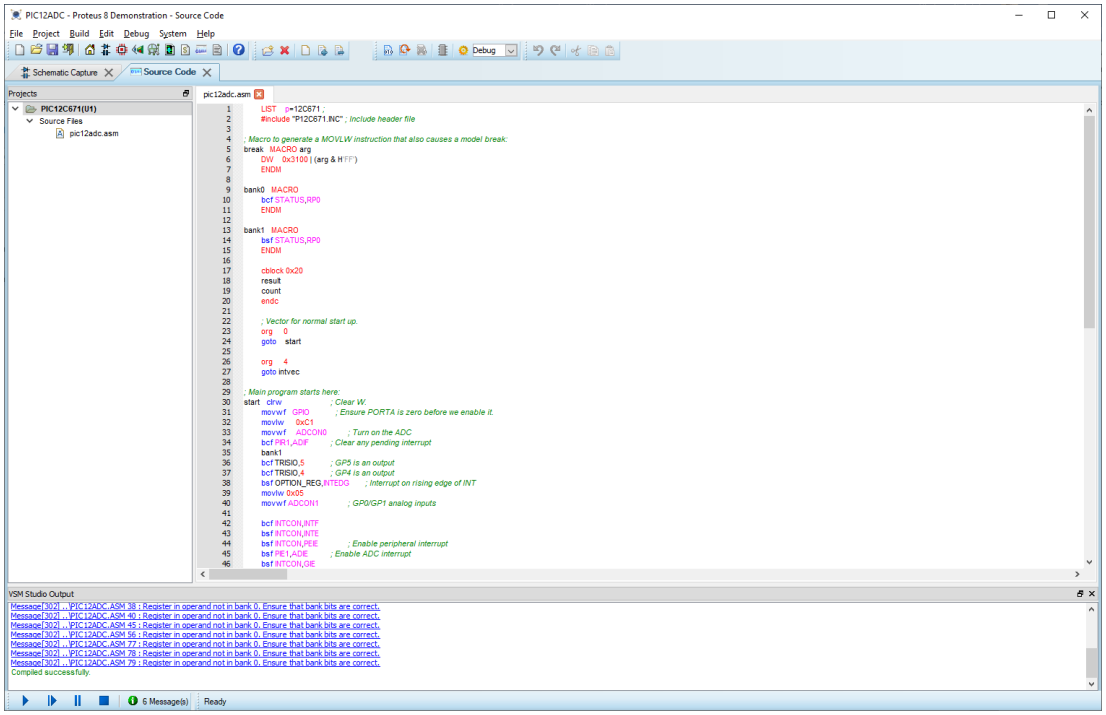

Рис. 4.4. Программа на ассемблере

И не только на ассемблере. Вы можете написать код программы на языке Си. Вот схема вольтметра, рис. 4.5.

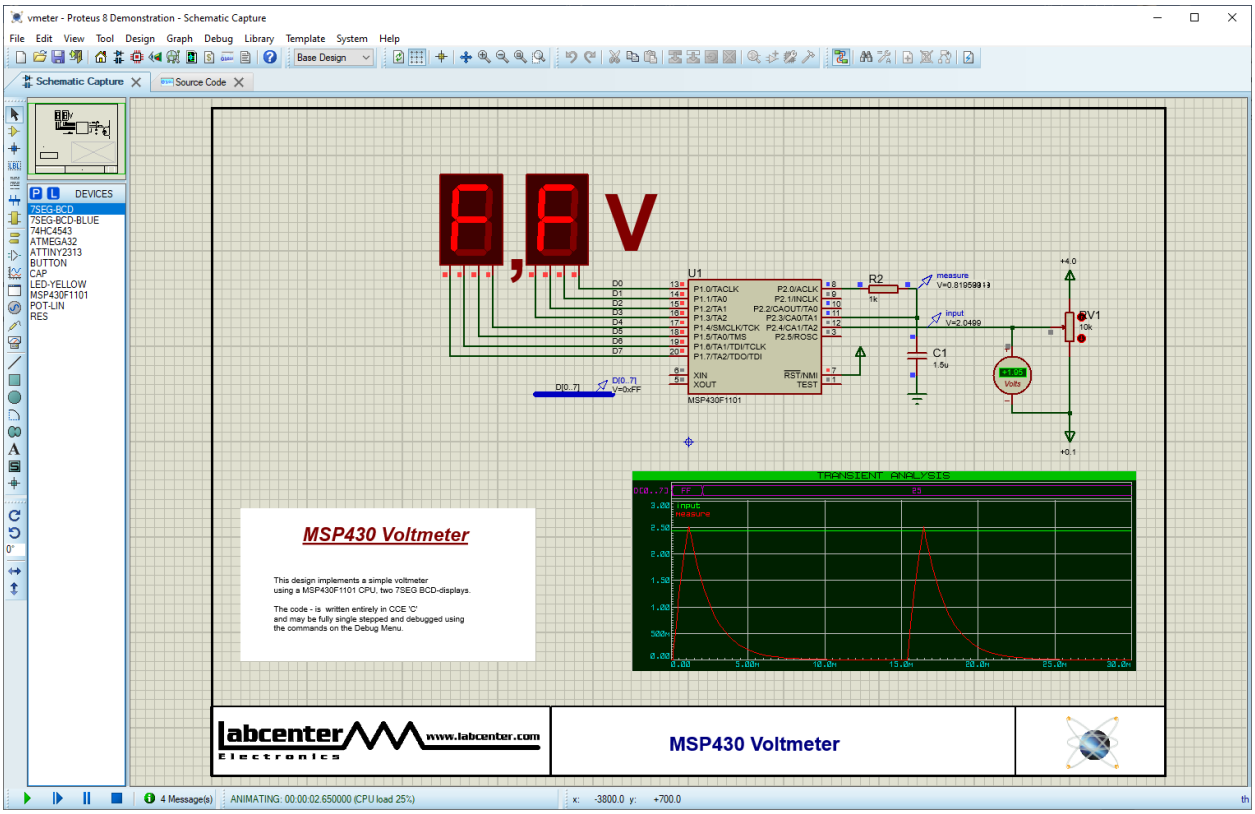

Рис. 4.5. Схема вольтметра

И вы можете убедиться, что программа написана на языке Си, рис. 4.6.

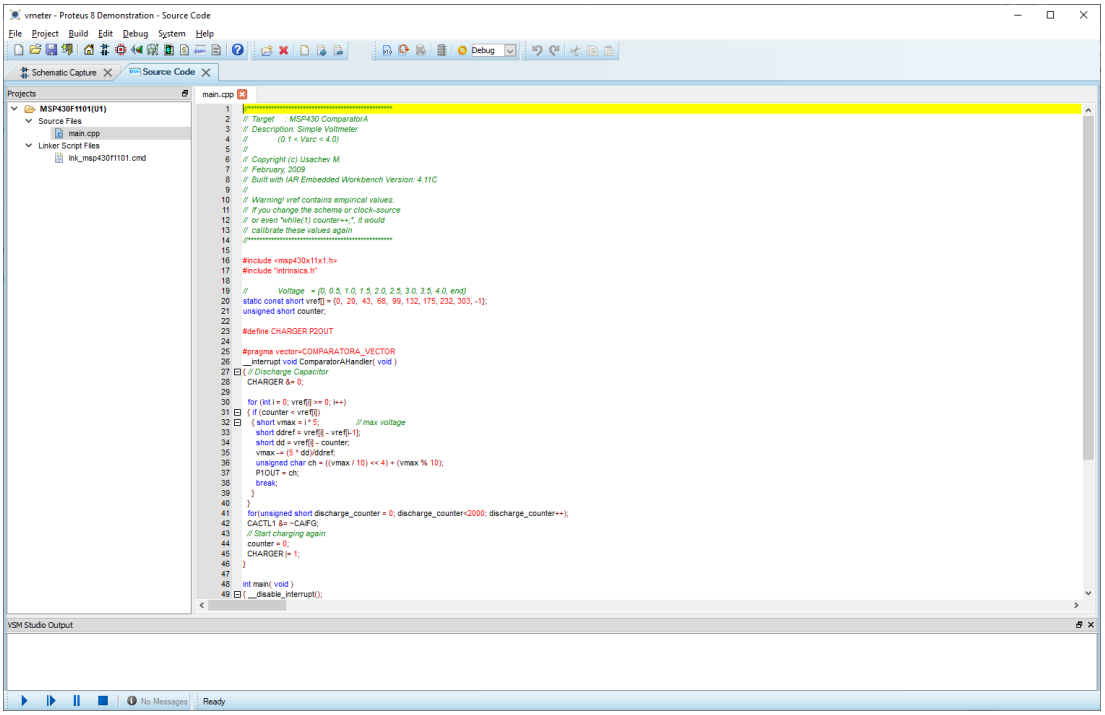

Рис. 4.6. Программа на языке Си

Что ещё привлекает внимание при беглом взгляде?

Конечно, привлекает внимание тот факт, что не только модули Arduino и модуль-компьютер Raspberry Pi доступны для работы с ними. Есть возможность использовать микроконтроллеры ARM. Если у вас есть модуль STM32F401, вы можете проверить, будет ли работать такая схема, рис. 5.1.

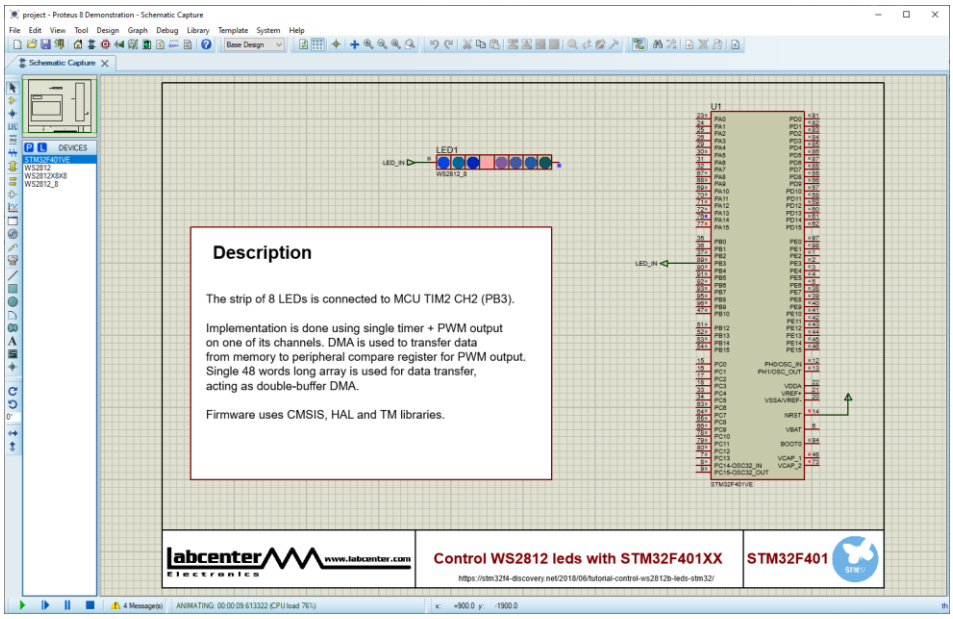

Рис. 5.1. Подборка светодиодов, управляемая предварительной записью в память

А как вам такой пример, рис. 5.2.

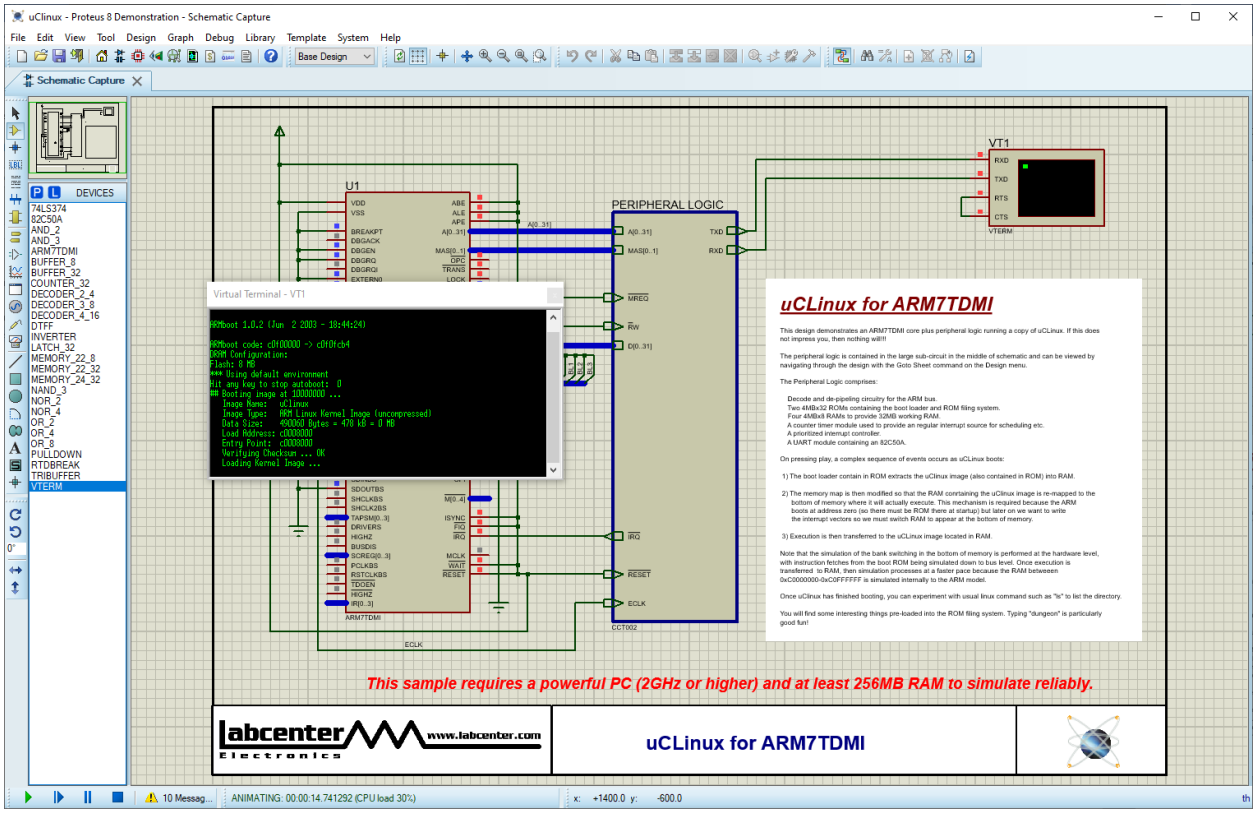

Рис. 5.2. Пример операционной системы на контроллере ARM

И вы можете проверить, что терминал (хотя бы в принципе) работает в Linux, рис. 5.3. Вот несколько команд, которые выполняются.

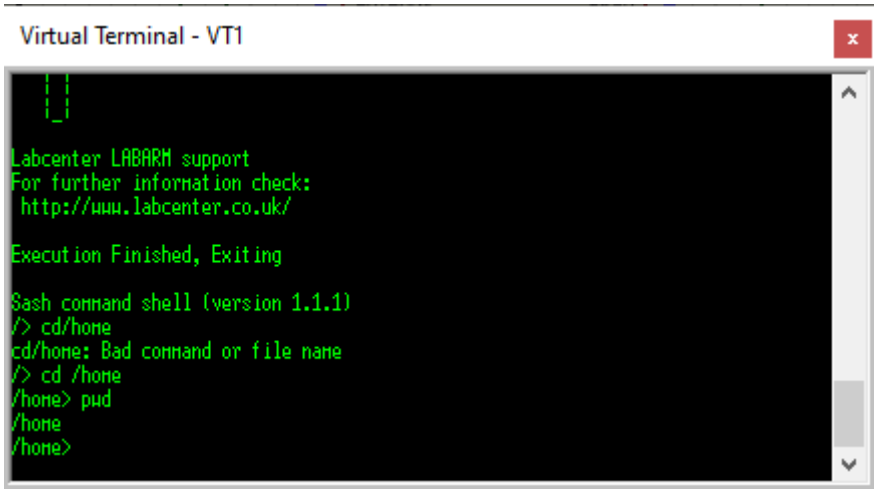

Рис. 5.3. Выполнение команд bash в Linux

Написать программу для STM32 вы можете на языке Си. Вот пример программы, написанной на языке Си, рис. 5.4.

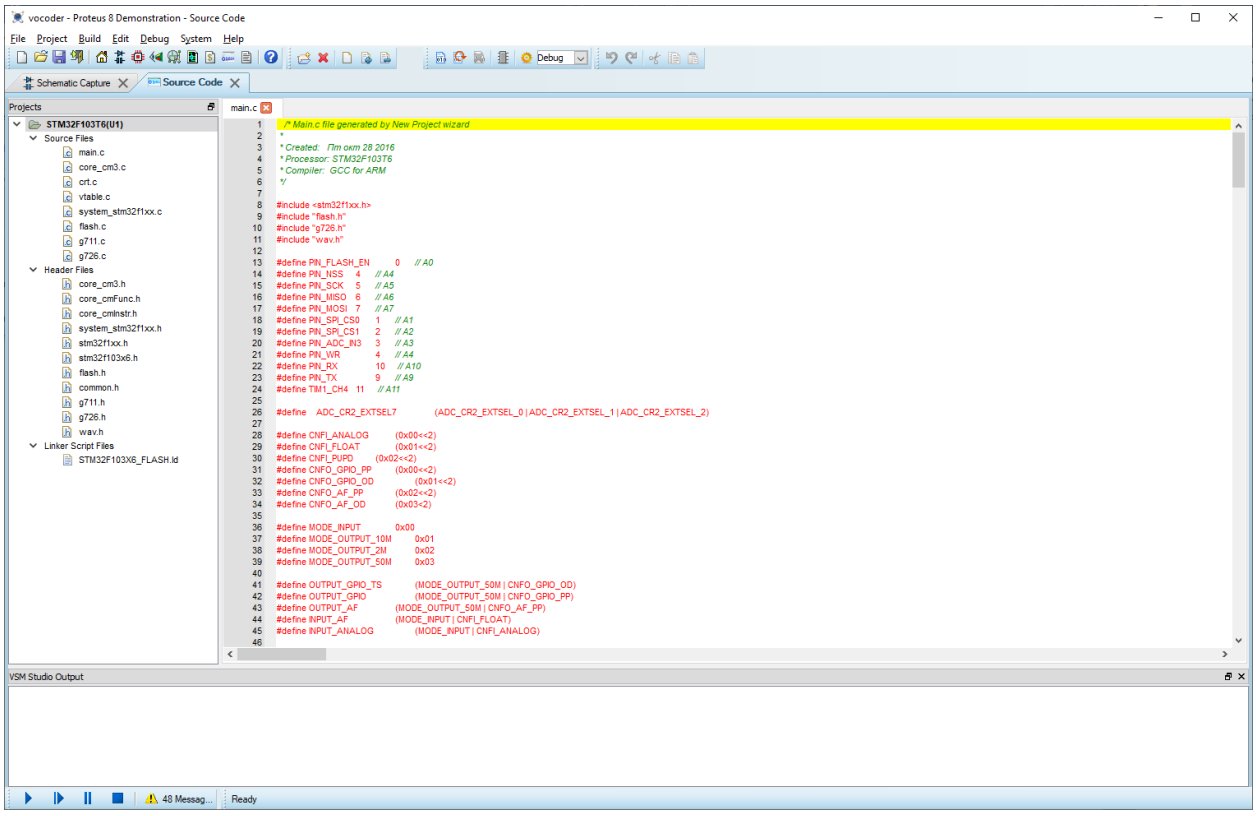

Рис. 5.4. Программа на Си для STM32

То есть, вы можете написать программу на Си, оттранслировать её и проверить её работу в режиме моделирования работы устройства, рис. 5.5.

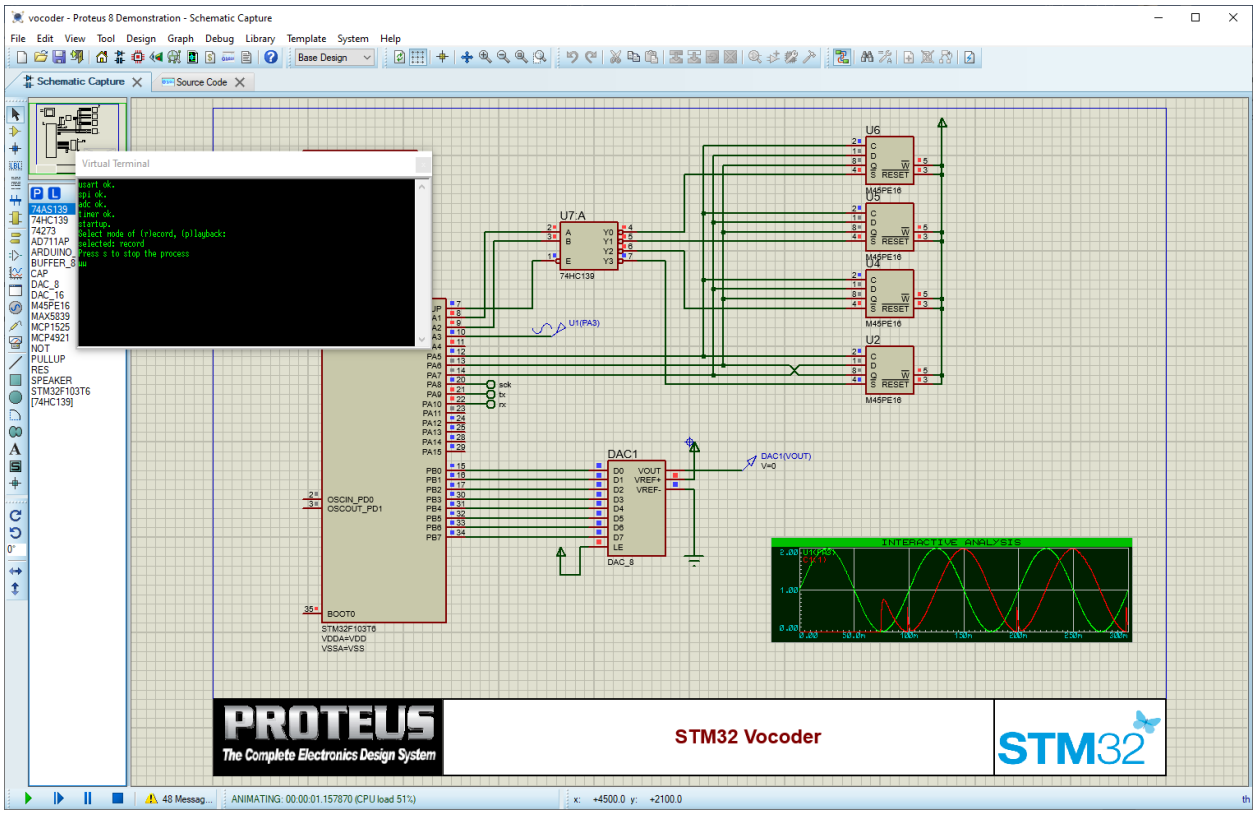

Рис. 5.5. Моделирование написанной программы

И последнее, что хотелось бы показать из более, чем 600 примеров, которые получаешь вместе с программой, рис. 5.6.

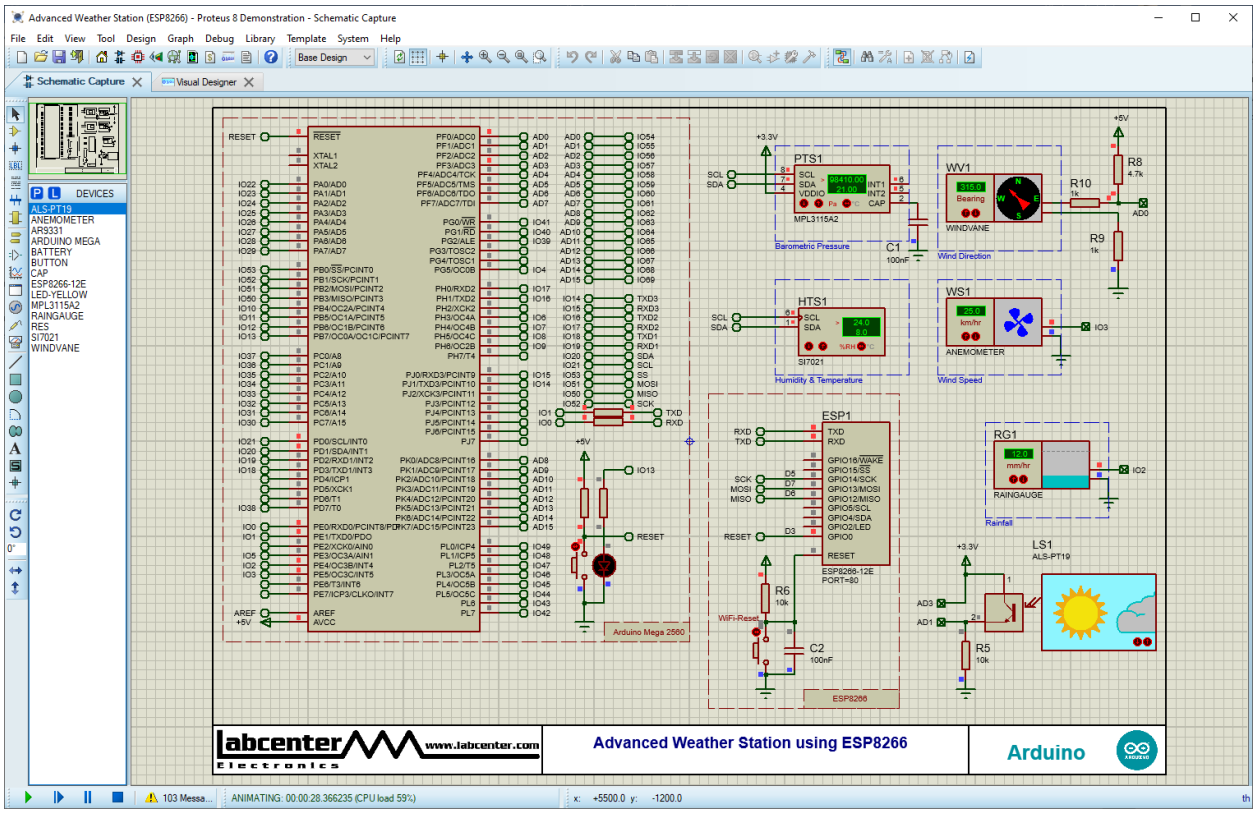

Рис. 5.6. Программа климат-контроля на базе модуля Arduino Mega

Эпилог

С момента предыдущего знакомства программа приобрела много новых возможностей. По словам разработчиков она находит широкое применение не только в сфере образования, но и в производстве. Что ж, с этим спорить трудно. Была бы она доступна в цене, её охотно применяли бы и радиолюбители. С чем тоже не поспоришь.