



ЗАМЕТКИ К ВЫХОДУ FLOWCODE 9



Vladimir Gololobov
МОСКВА - 2020

Оглавление

| | |
|--|----|
| Установка и начало работы | 2 |
| Начало работы над проектом..... | 5 |
| Шаблоны (шапочное знакомство с Templates) | 7 |
| Что не получилось в восьмой версии | 8 |
| Что нового в версии Flowcode 9 | 12 |
| Среда разработки дружелюбная к непрограммистам | 12 |
| Flowcode в производстве | 12 |
| Flowcode в образовании | 12 |
| Недорогие аппаратные интерфейсы | 12 |
| Так, что нового в девятой версии..... | 12 |
| Итог | 13 |
| App Developer..... | 15 |

Установка и начало работы

Необычной для меня оказалась установка программы. Понадобилось либо зарегистрироваться на сайте производителя, либо подтвердить регистрацию. Я помню пароли, явки? Пришлось вспоминать, чтобы заработала пробная версия.

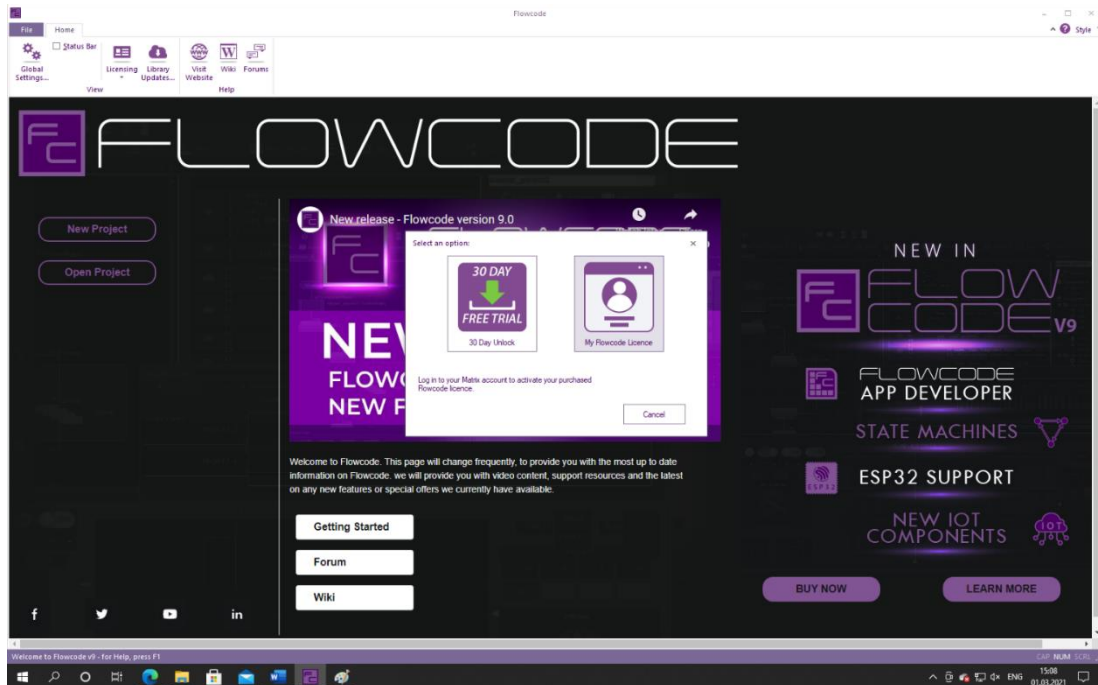


Рис. 1.1. Предложение использовать пробную версию или подтвердить лицензию

И, признаюсь, непривычным показался интерфейс программы. Не знаю, как работает купленный вариант программы, но пробная версия постоянно подгружает что-то. Что тоже непривычно. И, я не обнаруживаю привычной панели компонентов.

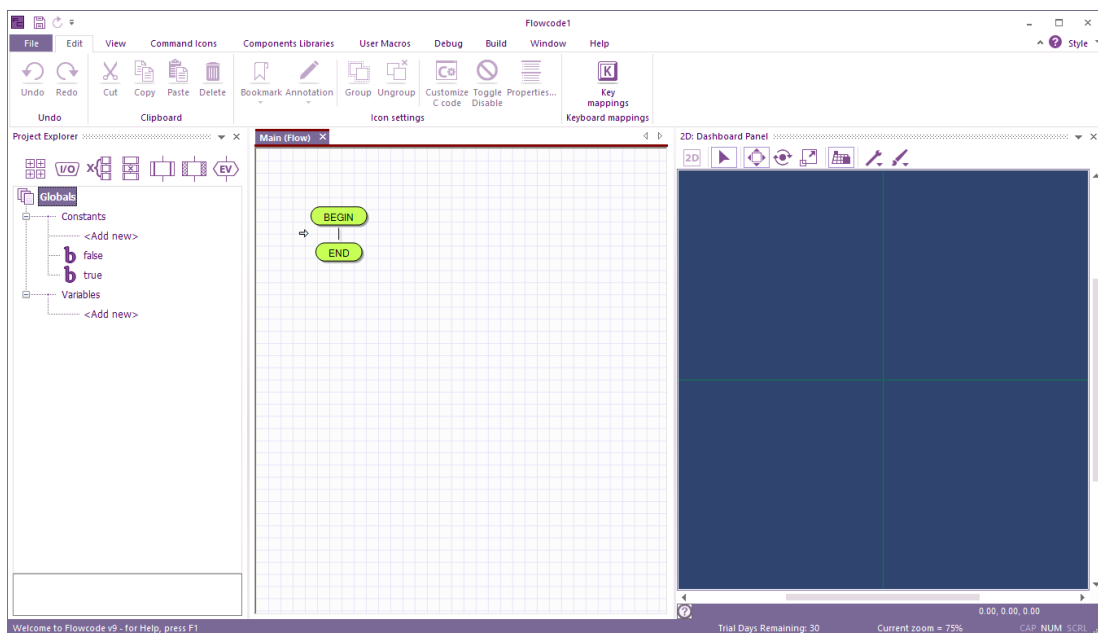


Рис. 1.2. Непривычный вид программы после создания нового проекта

Для тех, кто впервые знакомится с программой Flowcode, поясню, что, конечно, есть основное меню, есть панель инструментов; слева уже привычный для многих программ менеджер проекта, далее рабочее поле и следом панель для внешних элементов схемы. Всё так. И даже то, что я вижу в окошке менеджера проекта, мне знакомо – это диалог создания переменных и констант.

Однако ранее слева появлялась инструментальная панель компонентов – тех кирпичиков, из которых строилась сама программа. Впрочем, достаточно разобраться с этим, чтобы забыть о том, что было раньше. И начать всё с начала. Запускаем программу, в левой части окна выберем *New Project* (хотя полезно посмотреть видео, которое предлагает разработчик).

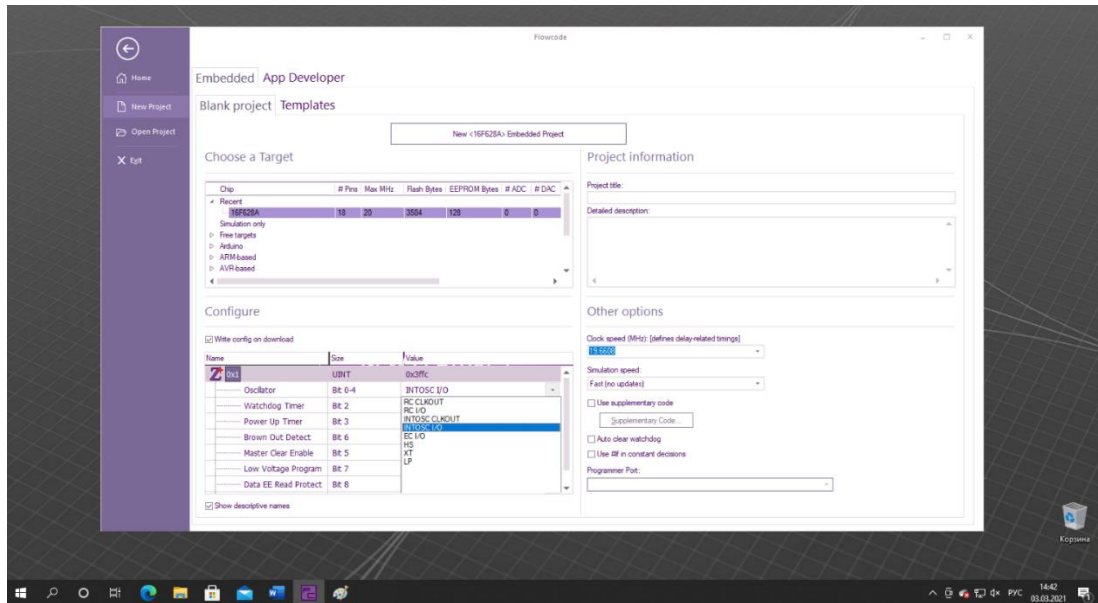


Рис. 1.3. Начало работы над проектом

Для первого проекта я выбрал PIC16F628A. Начинаящим проще поработать с этим очень полезным микроконтроллером. И они сразу могут задать слово конфигурации, что немаловажно, и указать тактовую частоту, что тоже важно.

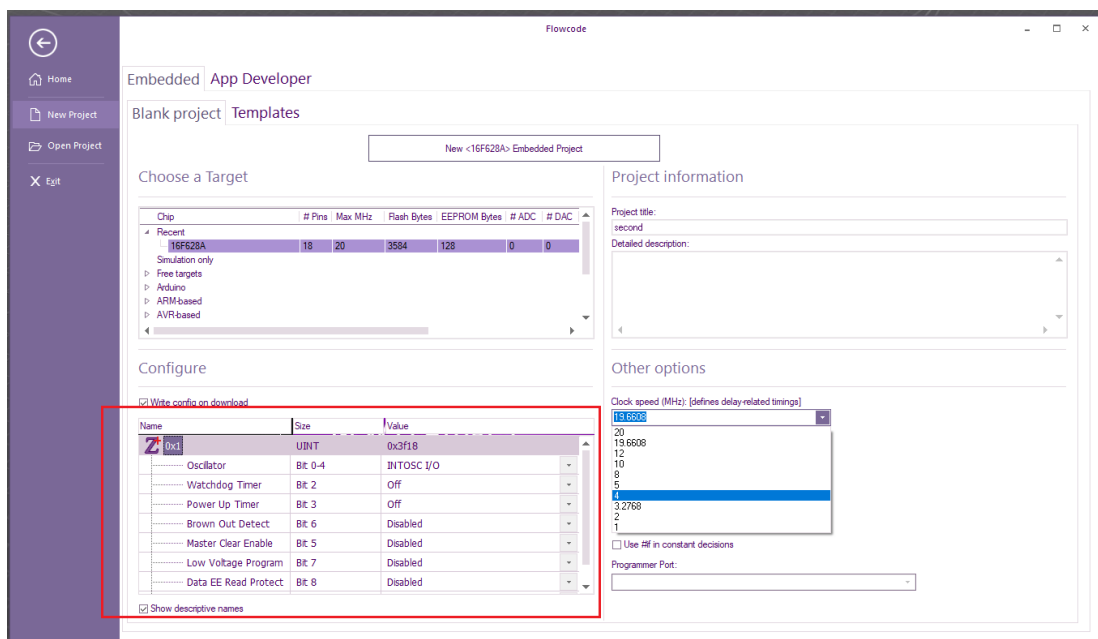


Рис. 1.4. Конфигурирование микроконтроллера до начала работы над проектом

Клавиша (или кнопка) с надписью *New <16F628A> Embedded Project* открывает рабочее окно. Оно может оказаться таким:

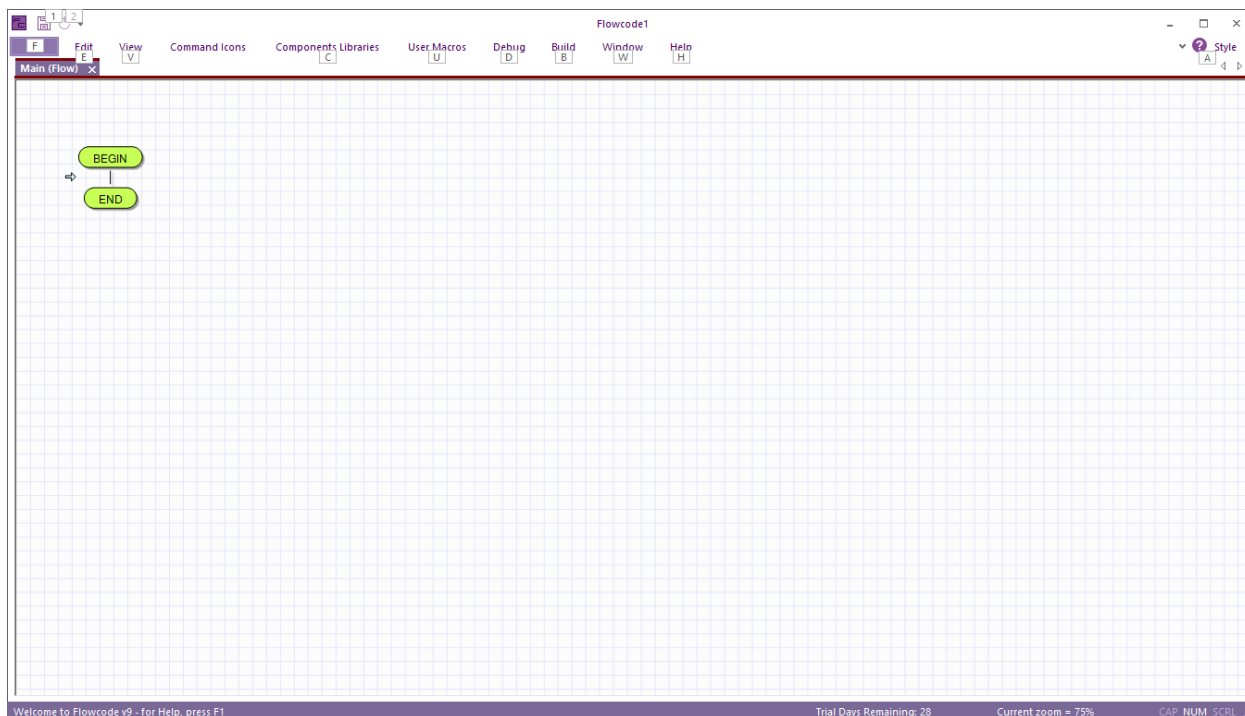


Рис. 1.5. Рабочее окно программы

Прежде, чем продолжить рассказ, замечу, что можно использовать готовые шаблоны для работы с разными модулями и микроконтроллерами. Например:

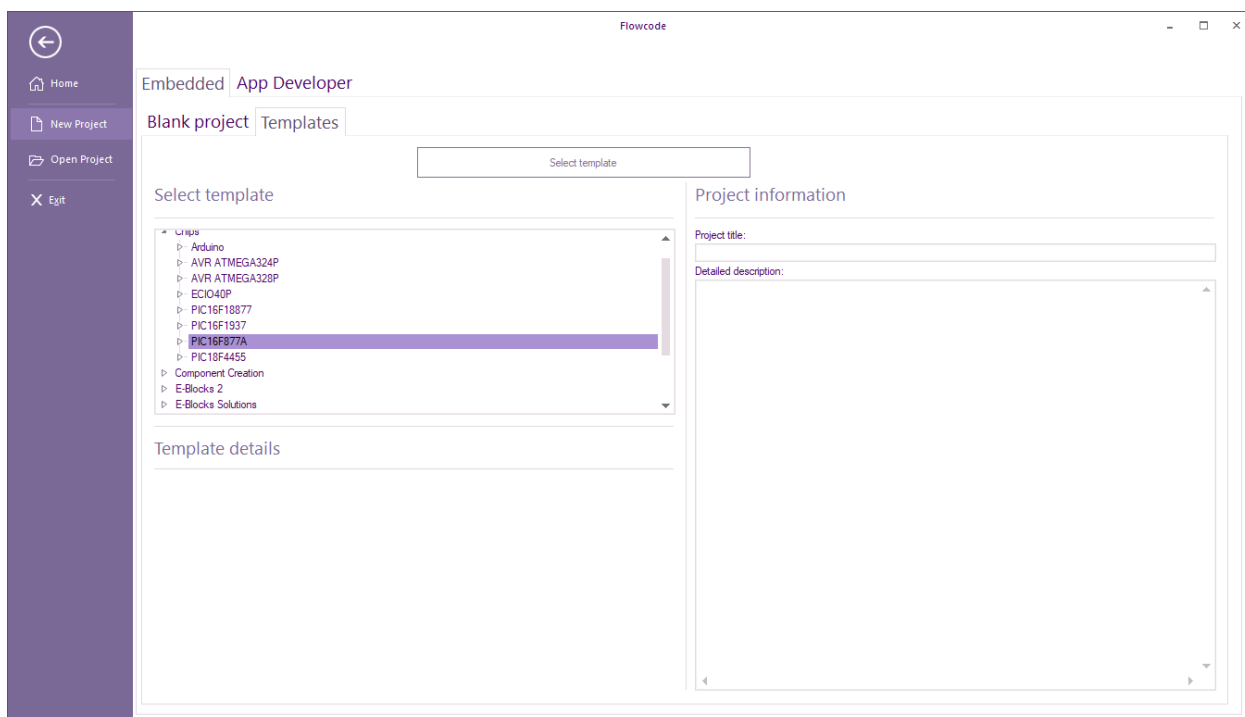


Рис. 1.6. Выбор готового шаблона для работы

Так многим начинающим можно посоветовать начать своё знакомство с микроконтроллерами, купив любой из модулей Arduino. Они сейчас недорогие, а сам проект очень удачен и продолжает развиваться.

Начало работы над проектом

Вновь, для тех, кто не знаком с программой, поясню, что программа собирается из графических компонентов, соответствующих стандартным языковым конструкциям. Так на рис. 1.5 есть начало и конец программы. Остальное следует заполнить вам. Для доступа к элементам программы, если у вас при входе в проект вид рабочего окна как на рис. 1.5, есть два пути: воспользоваться основным меню, где есть пункт *Command Icon*.

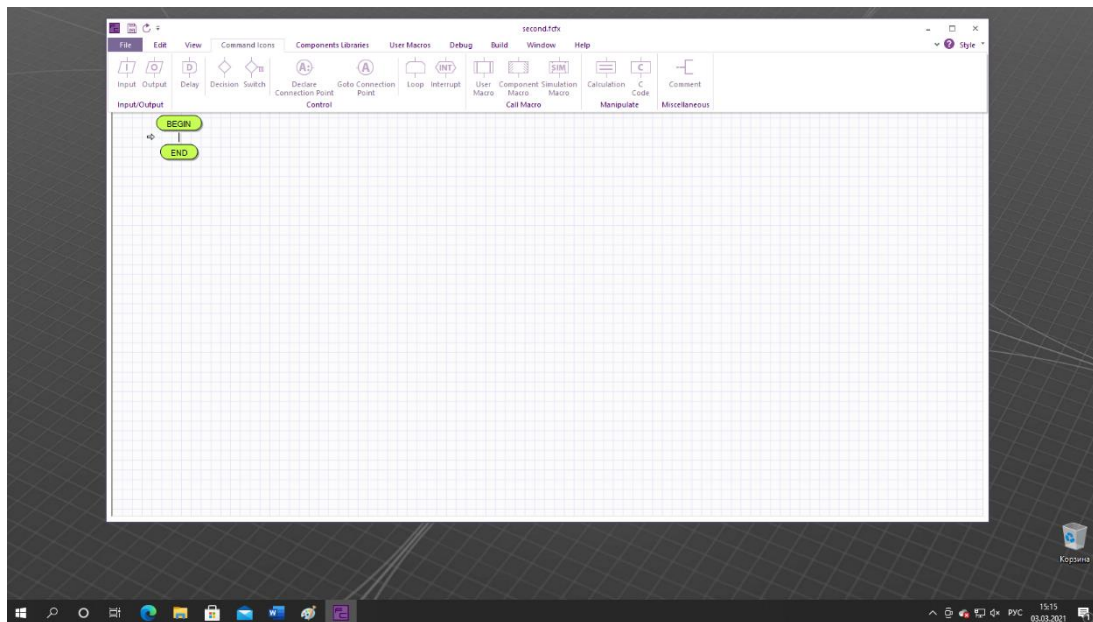


Рис. 2.1. Компоненты программы в основном меню

Можно поступить и иначе. В разделе основного меню View (вид) можно выбрать добавление к рабочему окну менеджера проекта.

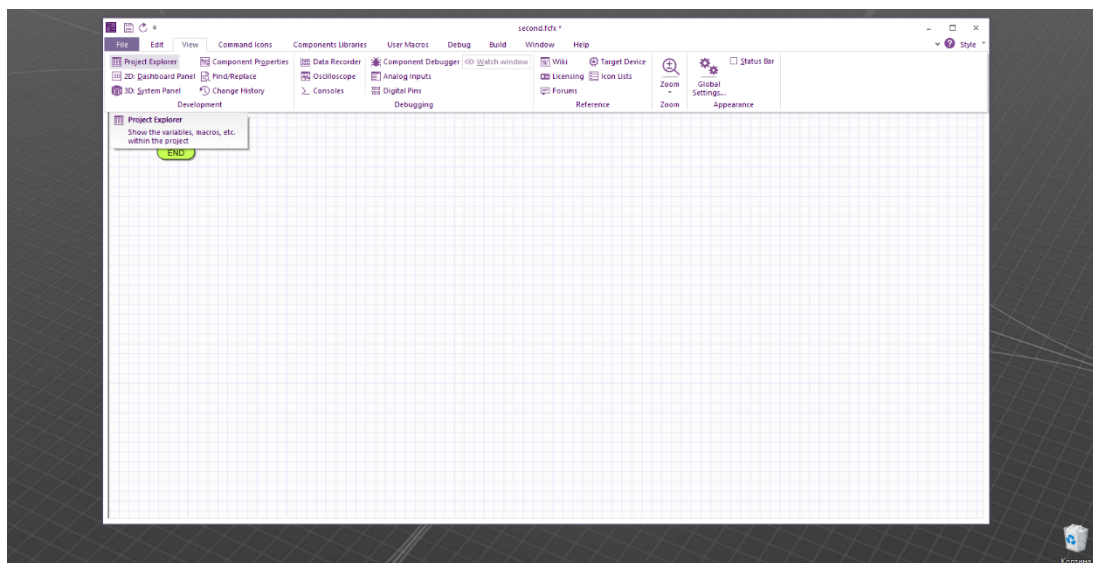


Рис. 2.2. Добавление окна менеджера проекта к основному окну

А после появления окна менеджера проекта использовать иконку (первую слева) для перехода к компонентам программы.

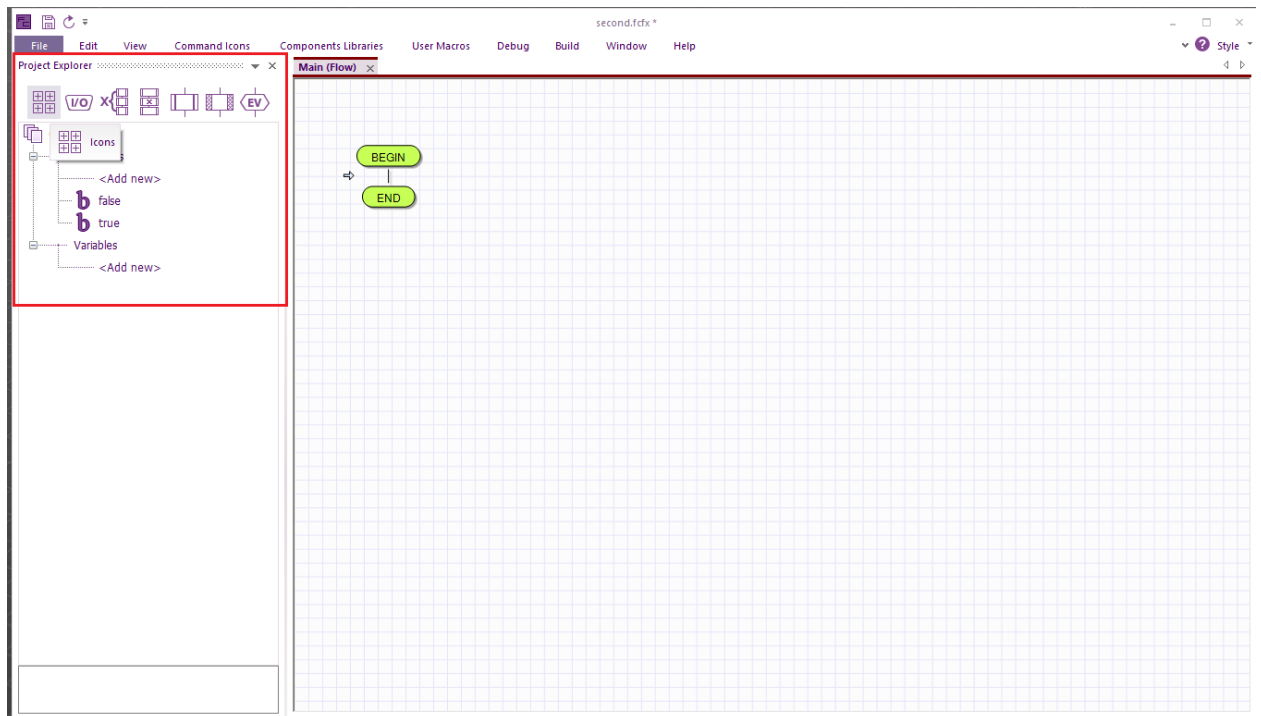


Рис. 2.3. Выбор показа компонентов в окне менеджера проекта

Теперь можно перетащить графические языковые элементы в программу. Нажимаем левую клавишу мышки, указателем выбираем нужный элемент, переносим его к линии, соединяющей программные скобки BEGIN-END, где его и оставляем.

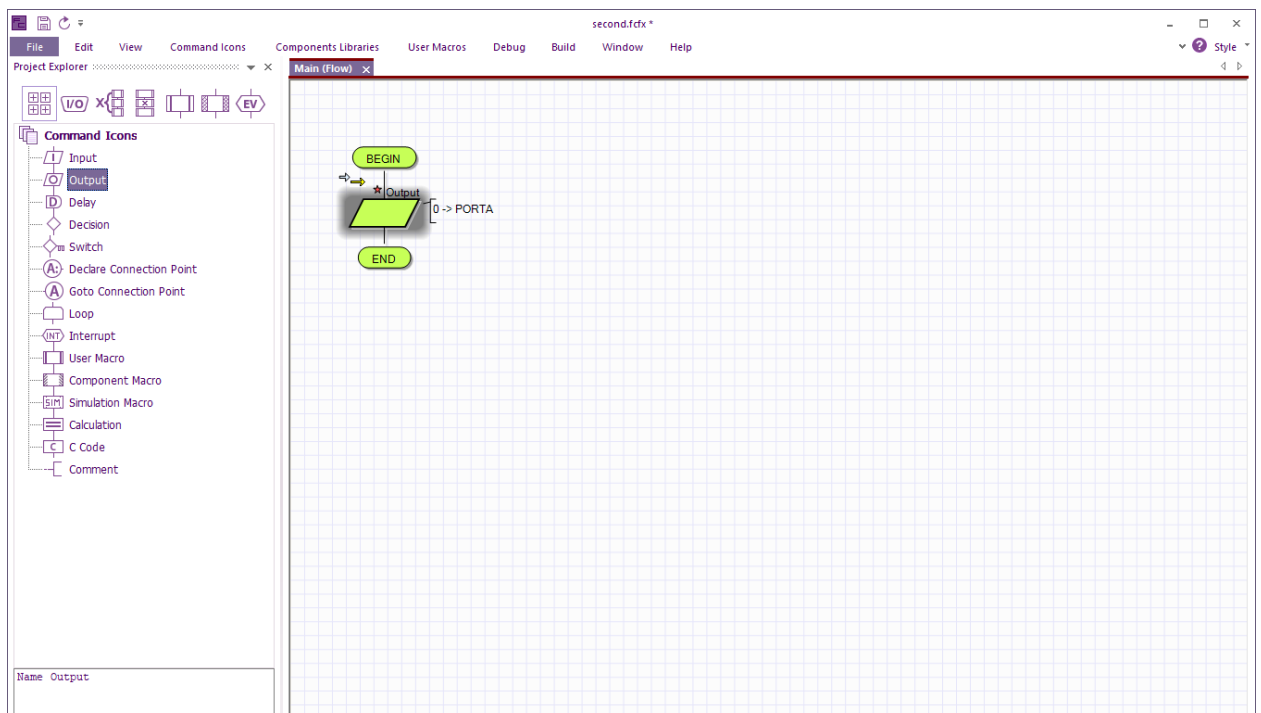


Рис. 2.4. Добавление компонентов программы

О том, как работать с программой можно прочитать или в моих рассказах, или в других книгах, руководствах. Я не стал бы останавливаться на таких простых вещах, если бы сам не был несколько обескуражен в начале знакомства.

Шаблоны (шапочное знакомство с Templates)

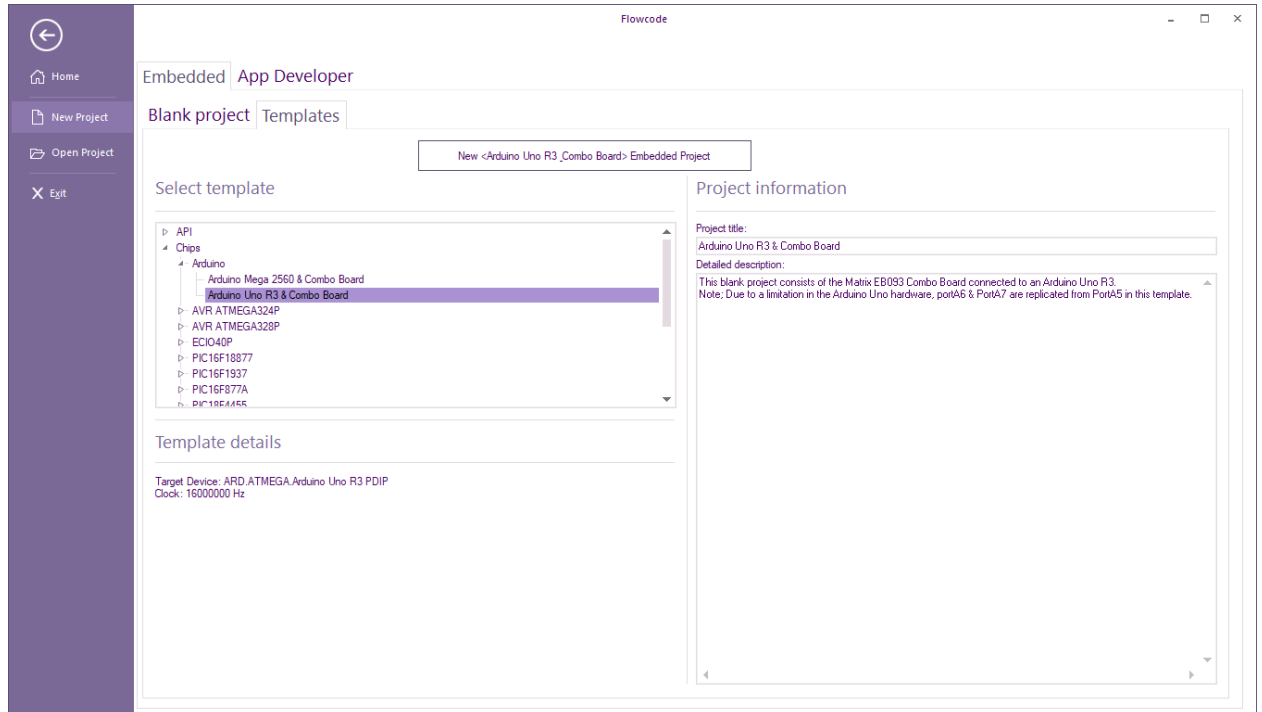


Рис. 3.1. Создание шаблона для работы с Arduino

Шаблон создаётся для дисплея, который я не могу найти в компонентах, которые можно добавить на панель. Поэтому я меняю его на дисплей из списка доступных. Добавление на панель LCD общего вида приводит к его появлению в компонентном макросе. Для него я и выбираю команду.

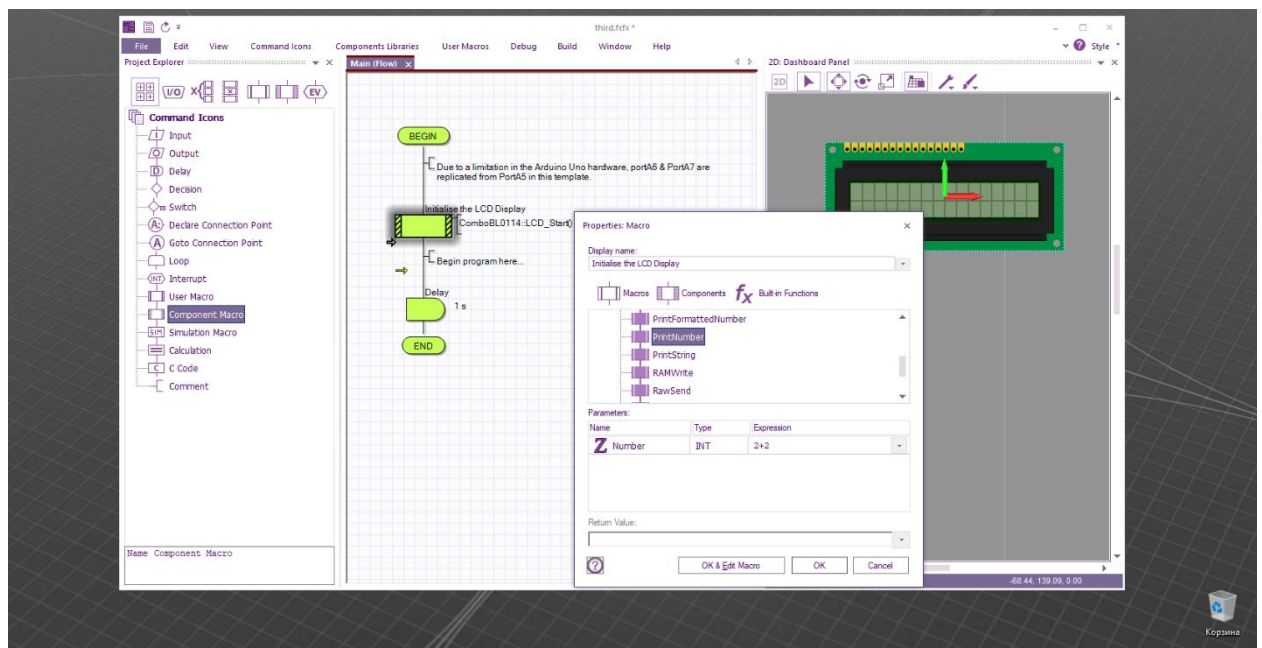


Рис. 3.2. Задание команды для вывода на дисплей

Сложить два и два не сверхзадача, но дисплей выводит 4, что свидетельствует – кто-то числа складывает, а дисплей это поддерживает. Я могу ошибаться, но шаблон, возможно, относится к комбинированному модулю на основе Arduino, как ряд других модулей, предлагаемых разработчиком программы. Если щёлкнуть по дисплею правой клавишей мышки, то в выпадающем меню можно выбрать свойства (Properties) и посмотреть подключение дисплея.

С шаблонами нужно разбираться, если это важно, если это нужно, если это интересно.

Что не получилось в восьмой версии

Рассказывая о восьмой версии программы, я обрадовался наличию осциллографа, но не смог его подключить, вышло отпущенное для знакомства время. Сейчас я хочу повторить работу с осциллографом, а попутно проверить, что со связью с предыдущей версией. Для этого загружаю программу, выбрав *Open Project*. В разделе вида выбираю осциллограф.

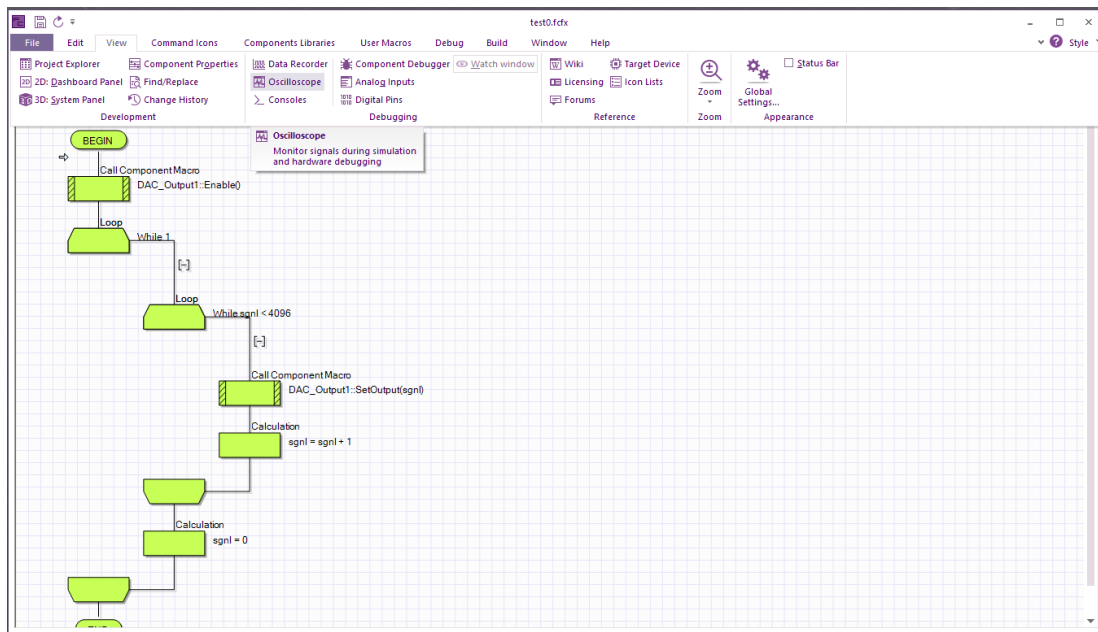


Рис. 4.1. Где находится осциллограф в программе Flowcode 9

Модуль STM32F407, для которого написана программа, имеет встроенный модуль ЦАП (DAC), который может генерировать (с помощью микроконтроллера) сигналы разной формы. После добавления осциллографа, нужно подключить его к выводу ЦАП. Добавляем «галочку» к первому каналу и нажимаем кнопку с надписью *Add data*. В диалоговом окне отмечаем, что сигнал аналоговый, можем назвать его, а к выбору вывода я не готов, оставляю An 0.

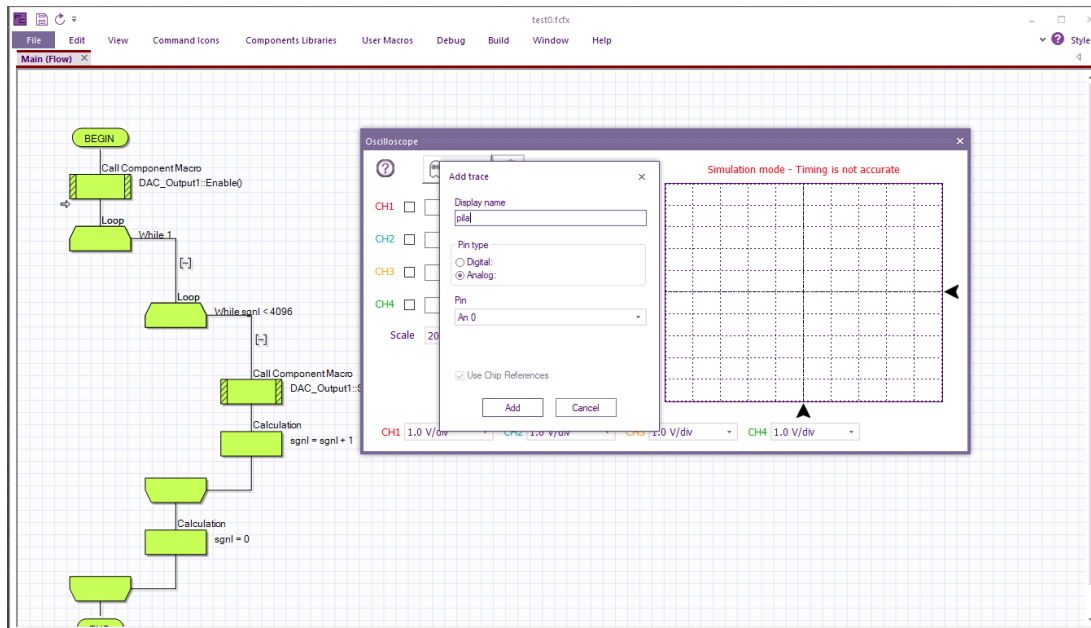


Рис. 4.2. Настройка осциллографа

Теперь выбираем в разделе Debug основного меню иконку Go. То есть, запускаем отладку программы.

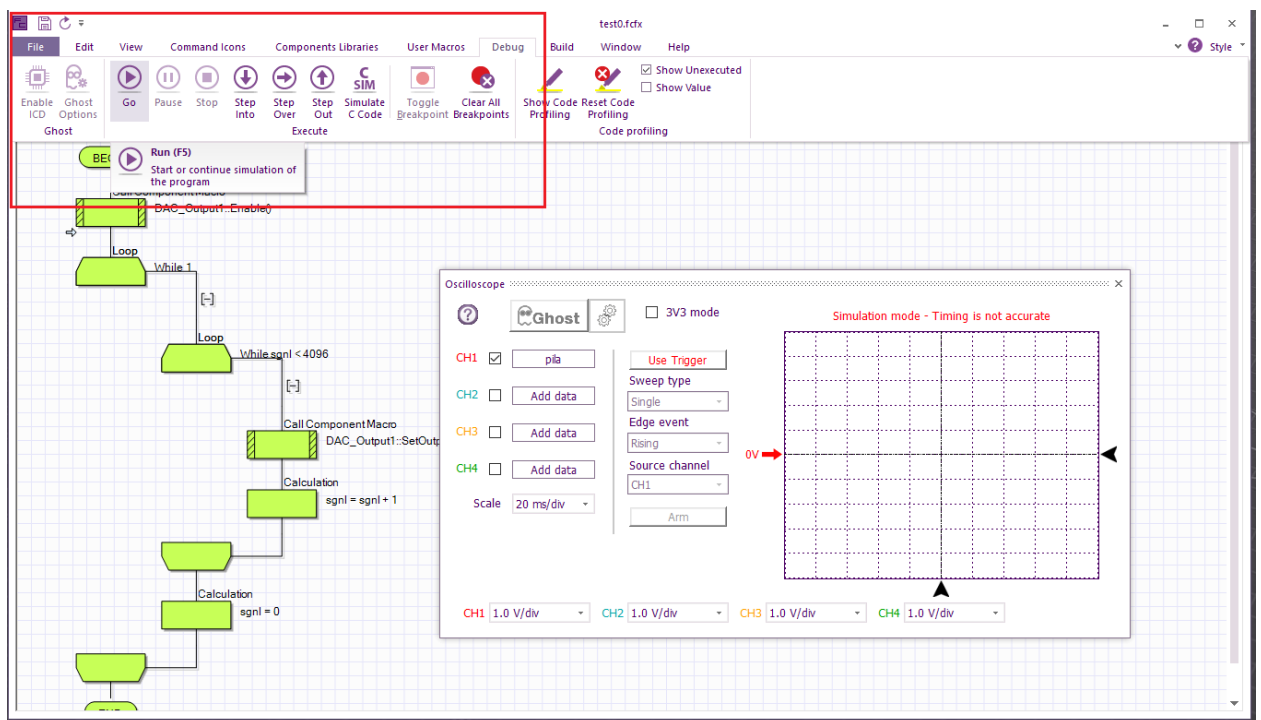


Рис. 4.3. Запуск программы

И в этот раз всё получилось. Мне не хочется возвращаться к восьмой версии, чтобы проверить, работает ли осциллограф так же, да и, скорее всего, у меня не получится повторить работу с пробной версией, лимит я исчерпал в прошлый раз. Будем считать, что всё бы работало.

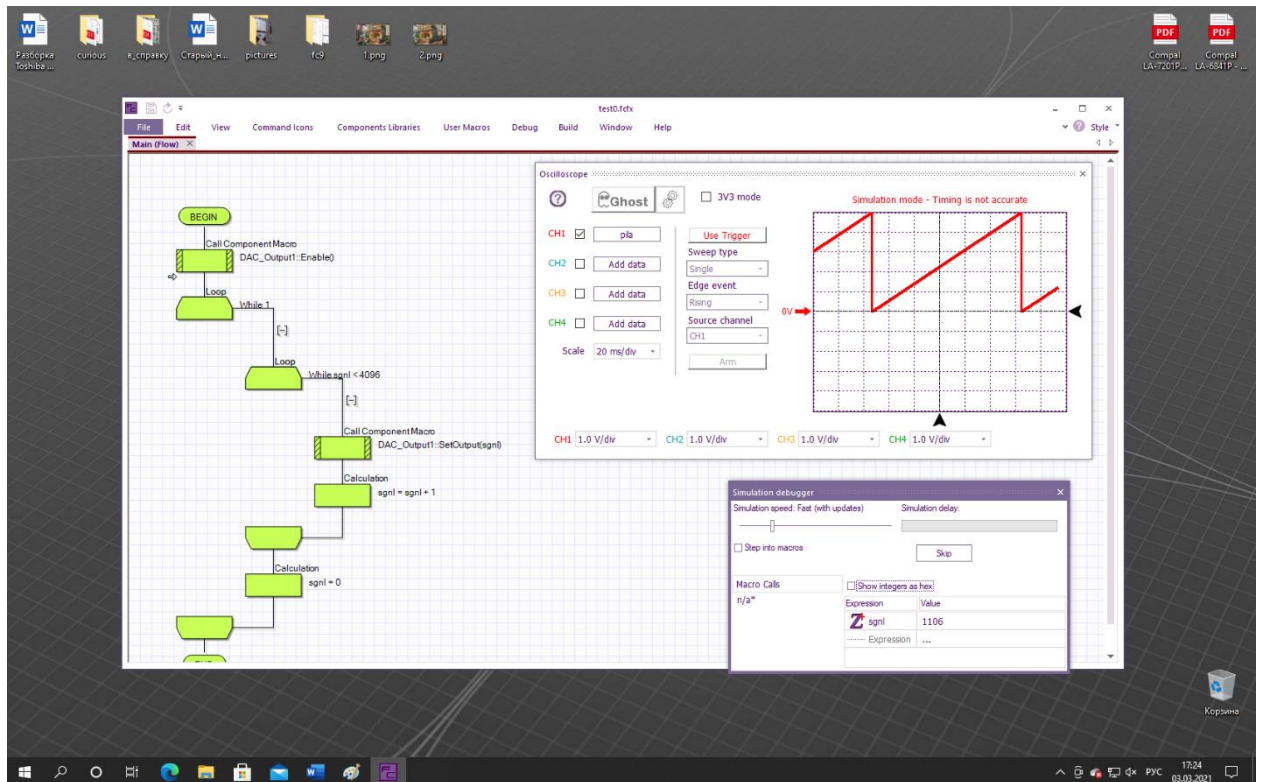


Рис. 4.4. Работа осциллографа с программой

В нижней части окна диалог, где можно менять скорость симуляции. Это очень удобно в тех случаях, когда важно пошаговое прохождение программы. Скорость можно уменьшить так, что программа будет медленно обходить все компоненты, а в окне наблюдения за переменными проверять правильность работы. В данном случае наблюдение за переменной *sgnl* показывает её изменение при проходе элементов программы по одному в секунду.

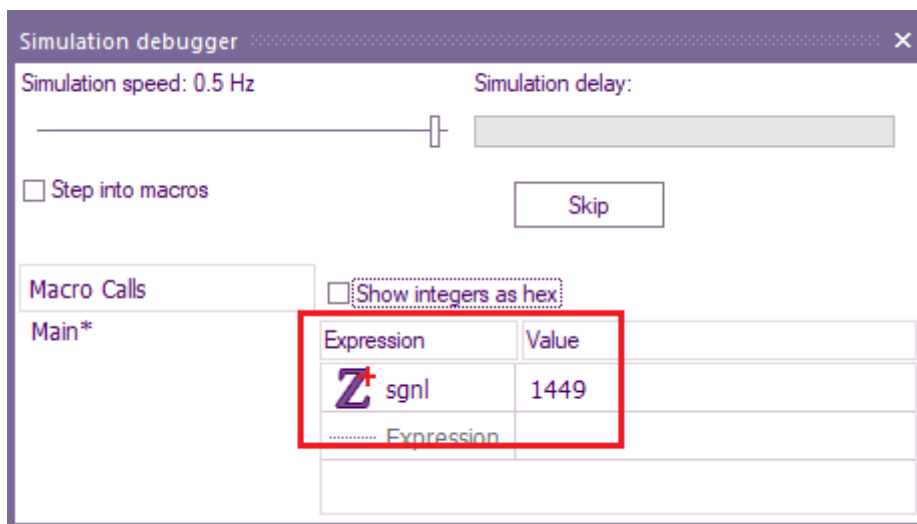


Рис. 4.5. Наблюдение за переменной

И завершить эту часть рассказа мне хотелось бы тем, что я заметил, когда начинал проект для PIC16F628A. В тот раз, оформляя данные проекта, мы настроили слово конфигурации. Для модуля STM32F407 настройки сложнее. Удобно, конечно, использовать STM32CubeMX в паре с удобной средой разработки. Но в среде разработки Flowcode тоже можно настроить конфигурацию контроллера.

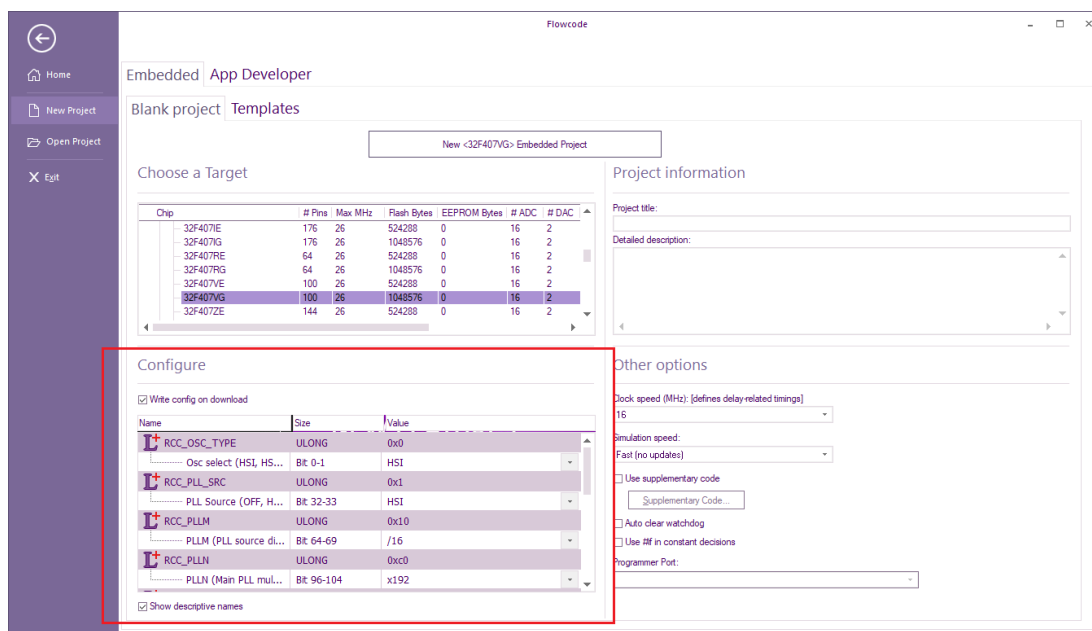


Рис. 4.6. Настройка конфигурации при работе с STM32F407

Пора, наверное, посмотреть, что пишут о программе и новой версии на сайте производителя.

Что нового в версии Flowcode 9

Начнём чтение с того, что такое Flowcode.

Flowcode – это интегрированная среда разработки (IDE) с графическим программированием, которая позволяет вам создавать высоко функциональные электрические, электронные и электромеханические системы на базе микроконтроллеров для ПК и планшетов с Windows.

Flowcode Embedded (встроенный блок-код) даёт возможность легко создавать программы с высочайшей функциональностью для популярных микроконтроллеров.

Flowcode App Developer (разработчик приложений Flowcode) позволяет вам создавать отличные интерфейсы для Windows ПК и планшетов; для взаимодействия человека с аппаратными системами и дешёвыми устройствами ввода-вывода; для локального взаимодействия через USB/Bluetooth или удалённо с помощью Wi-Fi, LAN, Internet.

Среда разработки дружелюбная к непрограммистам

Те, кто слегка знаком или не имеет опыта программирования могут беспрепятственно использовать Flowcode. Буквально за минуты начинающий может начать разрабатывать электронные системы, используя только пять доступных методов программирования.

Flowcode в производстве

Интуитивно понятная среда разработки Flowcode даёт пользователям возможность быстро создавать сложные системы. Промышленные инженеры используют Flowcode по той причине, что они экономят время, затрачиваемое на разработку, увеличивают функциональную надёжность; есть возможность разным группам кодировщиков работать над одним проектом, обеспечивая гибкость развертывания как на нескольких типах микроконтроллеров, так и на ПК.

Flowcode в образовании

Образовательные учреждения используют программу потому, что она доступна для всех инженерных специализаций студентов, включая электриков и механиков; тех, кто готовится к работе в аэрокосмической и автомобильной промышленности. Все инженеры нуждаются в понимании программирования, работы датчиков, исполнительных механизмов, систем обратной связи и управления. Среда разработки Flowcode позволяет начинающим инженерам стать успешными разработчиками систем независимо от их основной дисциплины.

Недорогие аппаратные интерфейсы

Flowcode Embedded создаёт hex-код для наиболее популярных во всем мире микроконтроллеров, включая ARM, PIC, Arduino, Raspberry Pi и ESP32. Пользователи могут использовать как популярные недорогие готовые платы, так и создавать свои собственные устройства. Код легко переносится с одного микроконтроллера на другой.

Так, что нового в девятой версии

State machines (конечные автоматы). Это новый графический интерфейс, который позволяет вам использовать эту мощную парадигму программирования. Присутствует во всех версиях.

App developer (разработчик приложений). Позволяет пользователям быстро и легко создавать программы для управления и наблюдения за внешними для ПК устройствами: на вашем столе, в вашем доме, в вашем городе или в другой стране.

ESP32 support (поддержка ESP32). Flowcode теперь поддерживает ряд новых устройств этого мощного 80 МГц уровня со встроенным стеком TCP/IP и встроенным Wi-Fi.

New IoT components (компоненты интернета вещей). Сейчас легко интегрировать ваши устройства в облачные платформы, используя Wi-Fi/LAN; можно использовать в качестве интерфейса человек-машина браузер, используя такие утилиты, как Thingspeak, iCloud и Mosquito.

Итог

Лучше один раз увидеть, чем сто раз услышать: [State diagrams - Flowcode Help](#)

На странице в Wiki есть примеры. Чтобы вас заинтересовать, я приведу один из примеров State diagrams.

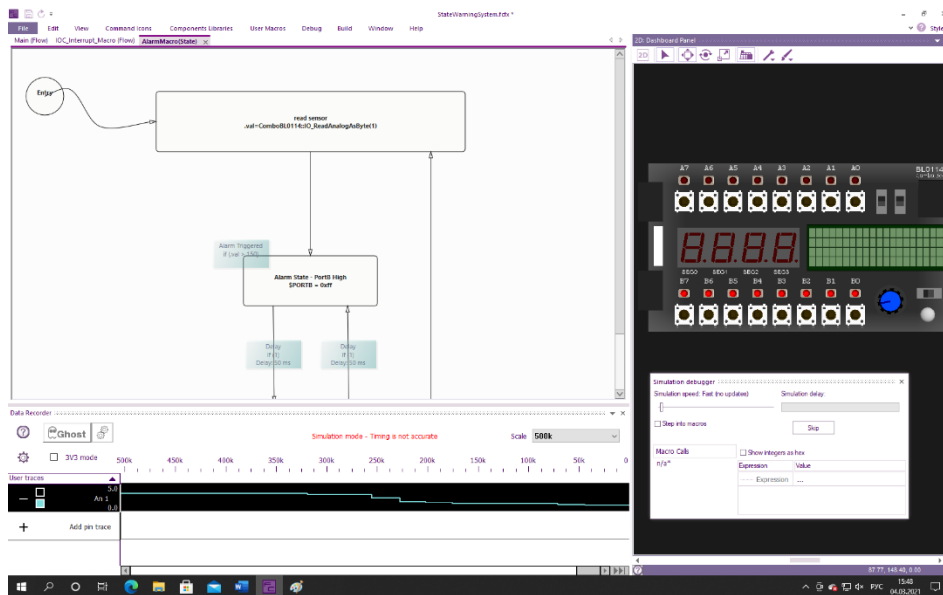


Рис. 5.1. Из примеров Flowcode на Wiki

Попутно, увидев воочию модуль, я вспоминаю, с чего начал описание шаблонов:

«Шаблон создаётся для дисплея, который я не могу найти в компонентах, которые можно добавить на панель».

Это, конечно, не дисплей, а модуль. Увидев его в приведённом выше примере, я решил вновь его отыскать. Перебирая компоненты в *Components Library*, я успеха не добился, но мог бы сообразить воспользоваться опцией поиска.

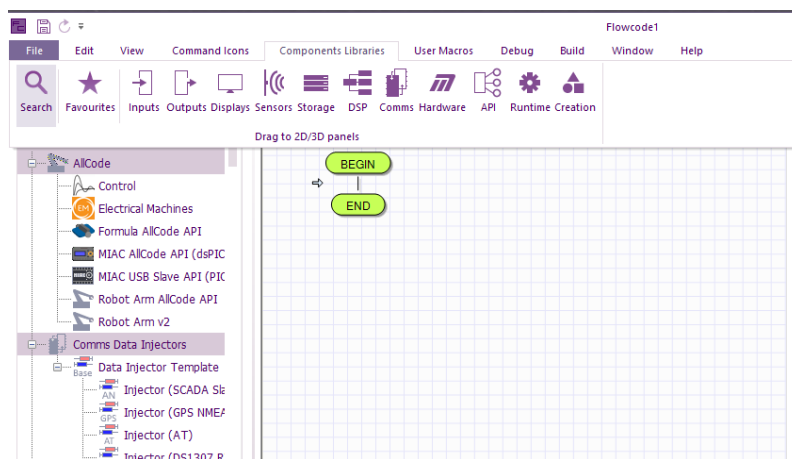


Рис. 5.2. Поиск в библиотеке компонентов

Здесь можно последовательно просмотреть все компоненты, а можно ввести в окно поиска нужный, чтобы сразу его найти.

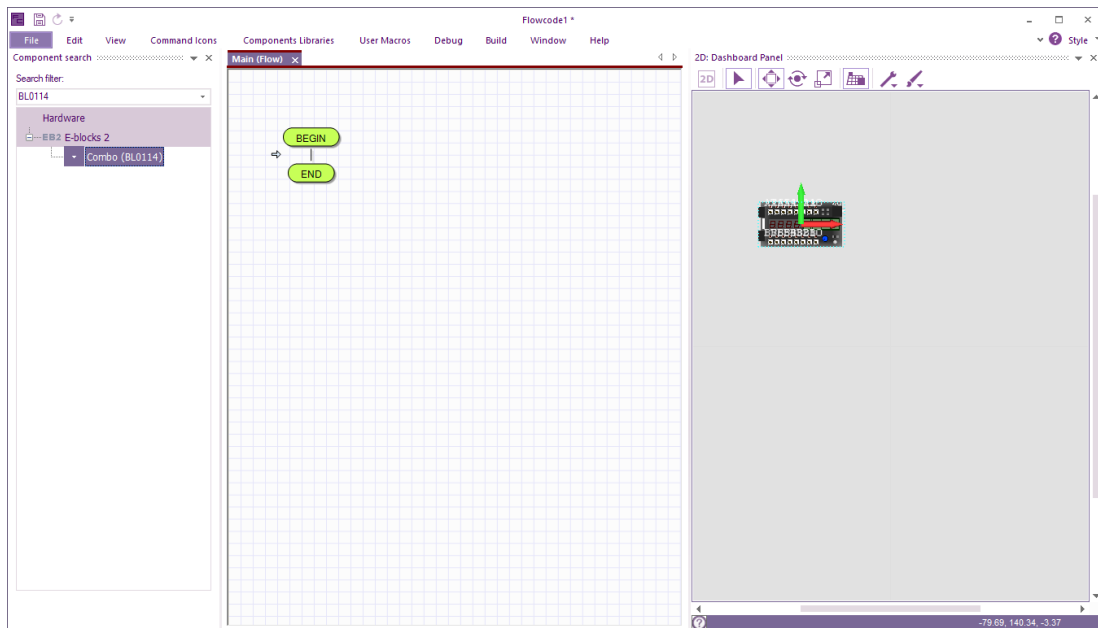


Рис. 5.3. Поиск компонента по его названию

Теперь с чистым сердцем можно повторить то, что я собирался сделать, но не сообразил, как...

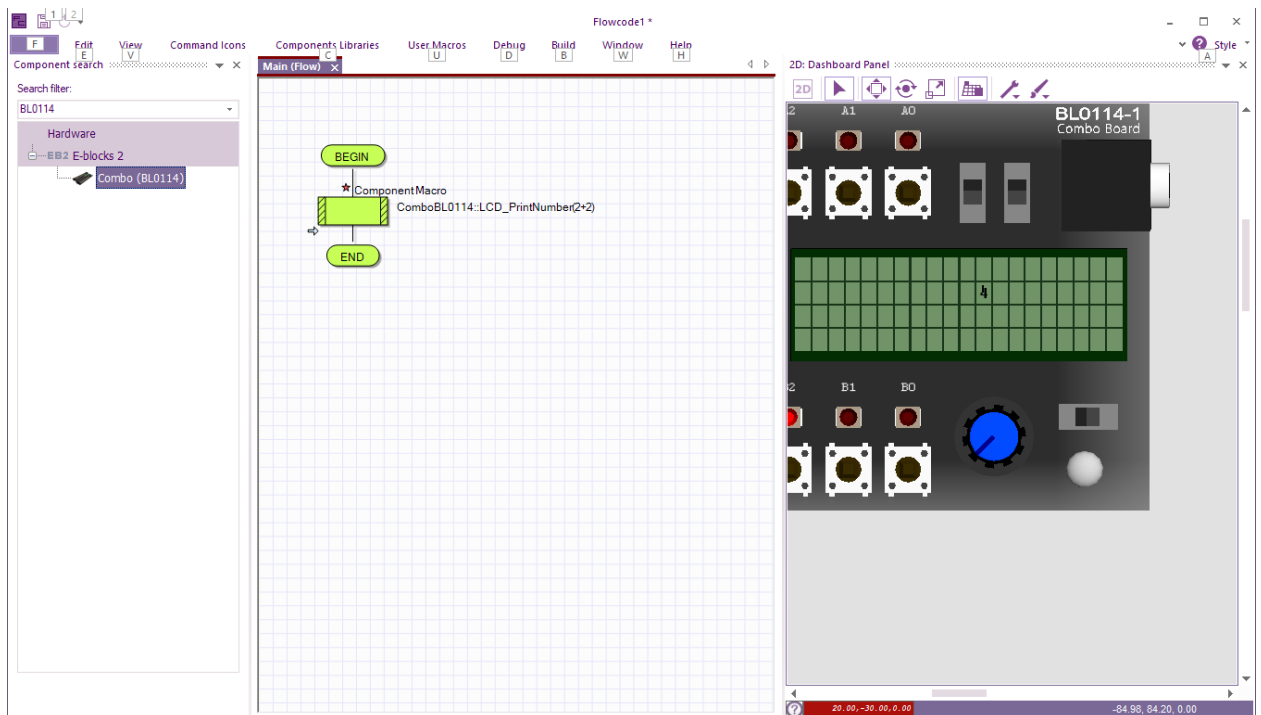


Рис. 5.4. Вывод на дисплей модуля BL0114

App Developer

Этот аспект работы в программе Flowcode вызывает у меня любопытство. А у вас?

Если тоже, то заглянем на страницу Wiki: [App Developer - Flowcode Help](#)

Когда я прочитал в разделе нового в версии Flowcode 9 об этой особенности, я подумал, что параллельно работе над программой для микроконтроллера можно создать и приложение, которое позволит общаться с микроконтроллером. Но так ли это?

Коли мы заглянули на страницу Wiki, считаем, что написано про *App Developer*.

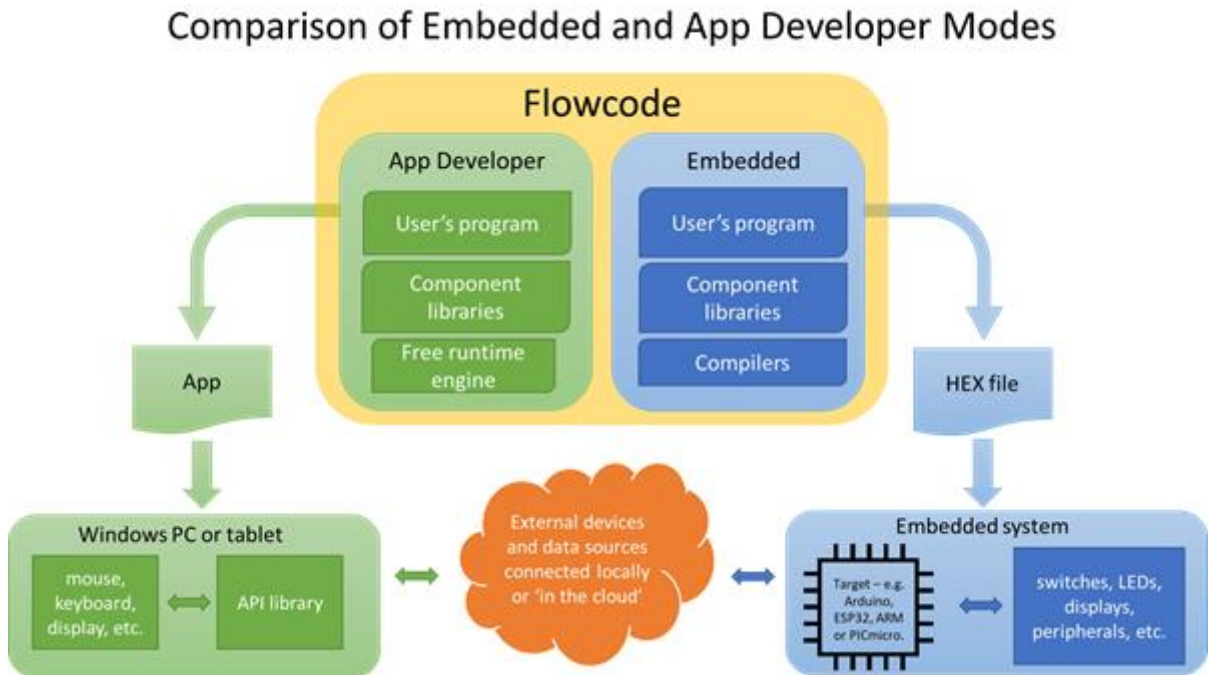


Рис. 6.1. Графика со страницы википедии

Под заголовком «Сравнение Встроенного и Разработчик приложения режимов» показана последовательность получения результата. В первом случае всё начинается с пользовательской программы, где добавляются компоненты из библиотеки, затем программа компилируется в hex-файл, который загружается в соответствующее устройство (в микроконтроллер устройства).

Во втором случае первые два этапа те же, но затем используется бесплатный движок для выполнения программы, что и даёт приложение, использующее компоненты ПК или планшета. Полученные результаты могут взаимодействовать через дополнительные устройства или через облако.

Если вам всё понятно, то мне не очень. То есть, я понимаю, как получается работающая программа для микроконтроллера, но не понятно, как создаётся приложение, увы. Вот, что написано далее.

В восьмой версии Flowcode мы ввели совершенно новую функциональность, которую назвали «SCADA». В девятой версии мы решили переименовать её в «App Developer», поскольку прежнее название не раскрывало её полного потенциала. App Developer позволяет вам легко создавать программы для управления и наблюдения за внешними устройствами по отношению к ПК. Но и это не всё. Flowcode App Developer позволяет вам создавать программы общего назначения с интерфейсом пользователя, который рассчитывает, строит и отображает информацию. Он

может получать информацию информация для вашего рабочего стола, из вашего дома, в вашем городе и по всему миру через Интернет.

Flowcode App Developer использует тот же интерфейс, с которым вы работаете при программировании микроконтроллеров. То есть, если вы знакомы с Flowcode, вы можете легко разработать приложение для Windows для собственных нужд или для безвозмездного распространения среди других.

Одной из особых сильных сторон является то, что Flowcode App Developer содержит большое количество коммуникационных компонентов, позволяющих создавать распределенные системы управления, сбора и анализа данных, поэтому он особенно хорош при создании SCADA-подобных систем с использованием интерфейсов прикладного программирования (API). Поддерживает популярные USB, Bluetooth и веб-аппаратные платформы подобные Modbus PLCs, Arduino, Raspberry Pi, Ethercat и др.

Flowcode App Developer – это платное программное приложение. Оно позволяет клиентам создавать бесплатные приложения, которыми можно поделиться со многими пользователями

Как по мне, то самое интересное здесь примеры. Постараюсь выбрать один несложный, но дающий некоторое представление о том, «с чем это едят».

Из представленных примеров выбран Parametric_graphs_test.fcsx. Выбран по причине того, что не требует подключения внешних устройств. Есть и другие, впрочем, причины. Итак. Открываем этот проект в Flowcode 9.

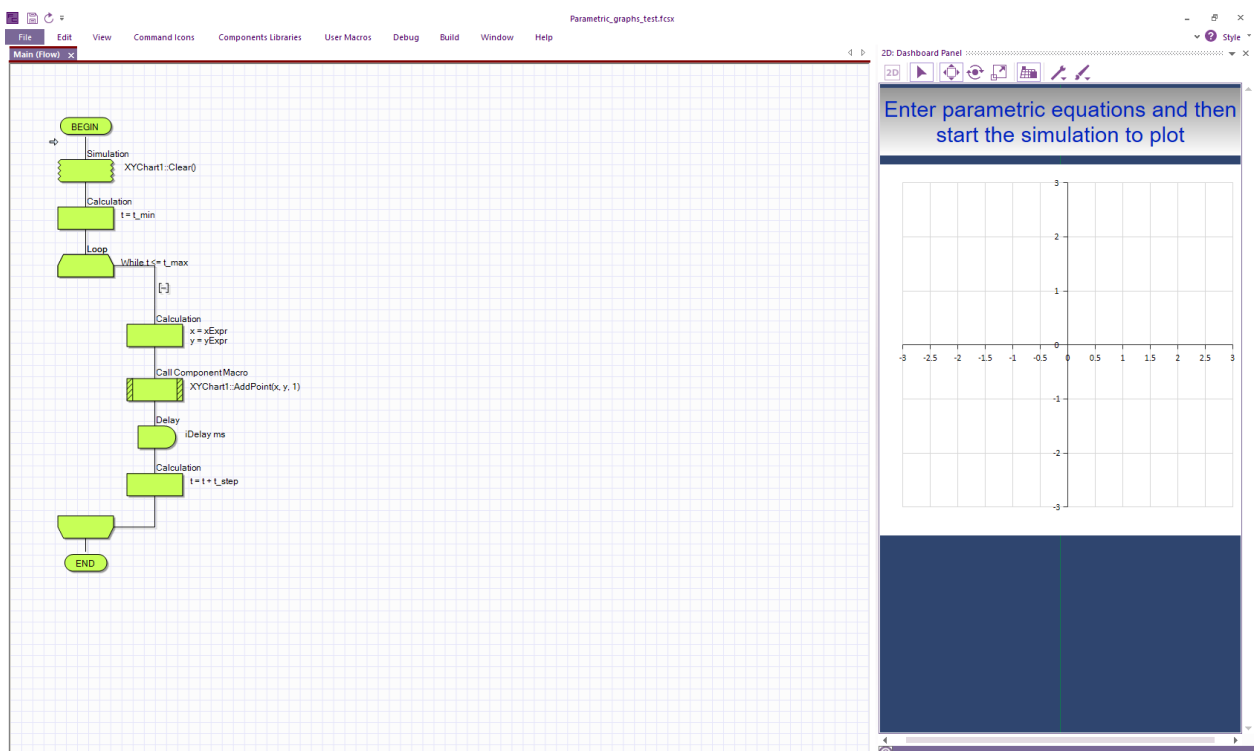


Рис. 6.2. Пример построения графика

Используем раздел *Debug* основного меню для запуска симуляции. Запуск иконкой с надписью *Go* и изображением привычной кнопки пуска.

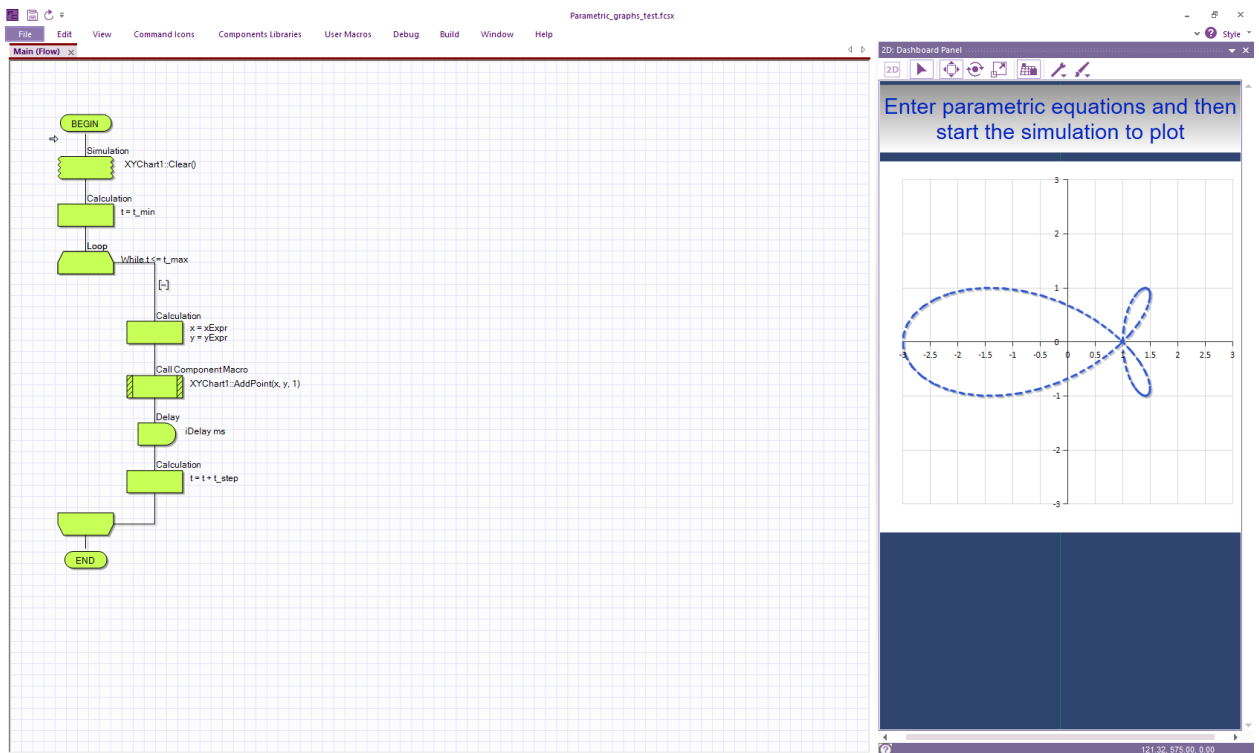


Рис. 6.3. Выполнение программы

В разделе помощи есть предложение что-то сделать, что я не совсем понял. Однако в разделе основного меню *File* есть возможность экспорта.

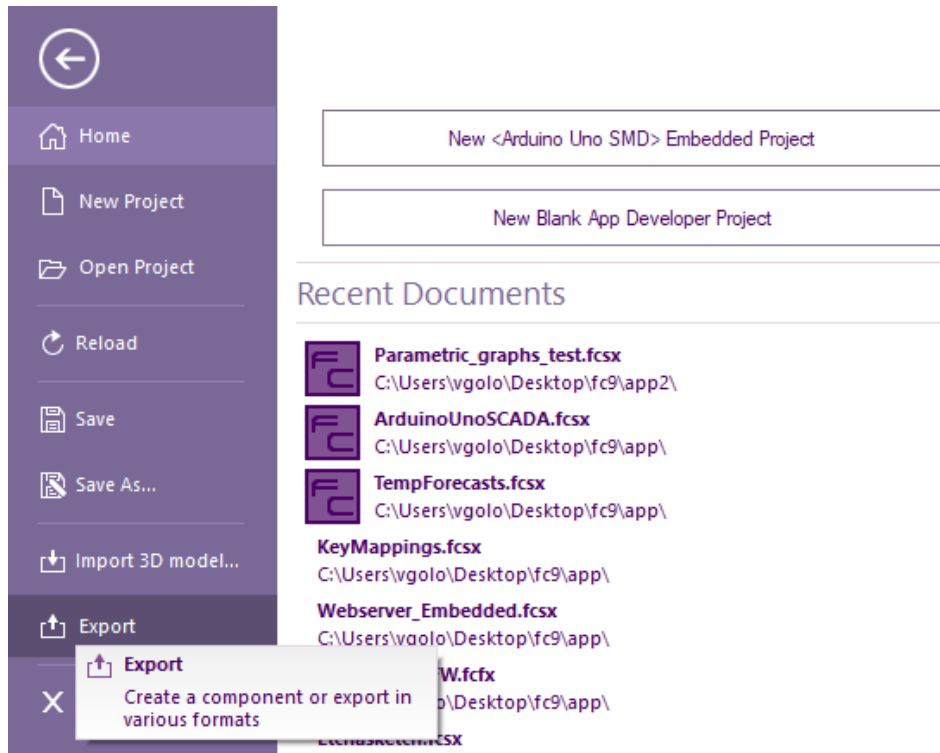


Рис. 6.4. Раздел File основного меню

Заглянем в подраздел Export.

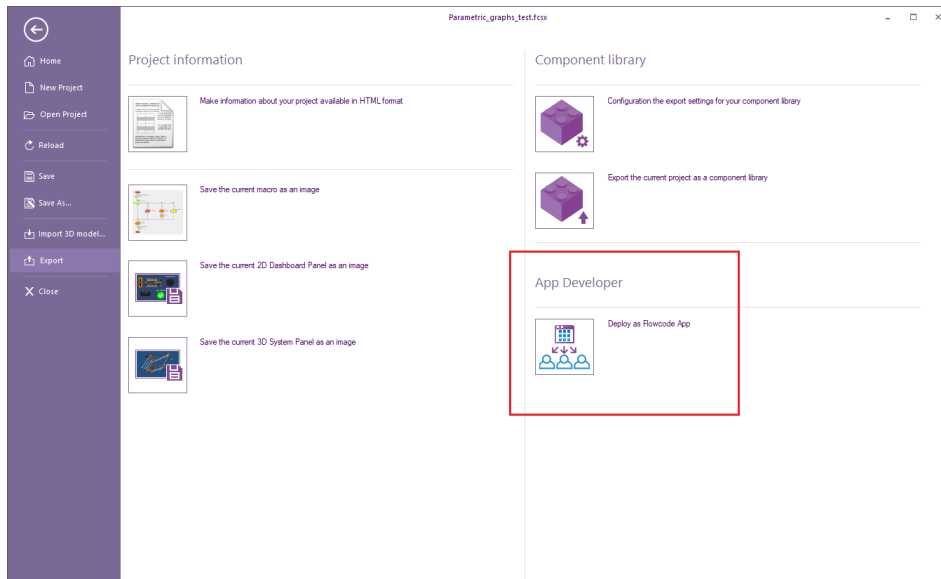


Рис. 6.5. Что можно экспортировать

В данном случае меня интересует активированный в этот раз App Developer (я пробовал другой проект, там окно оставалось неактивным). Попадаем в диалог настройки экспорта, который пока оставим без изменений.

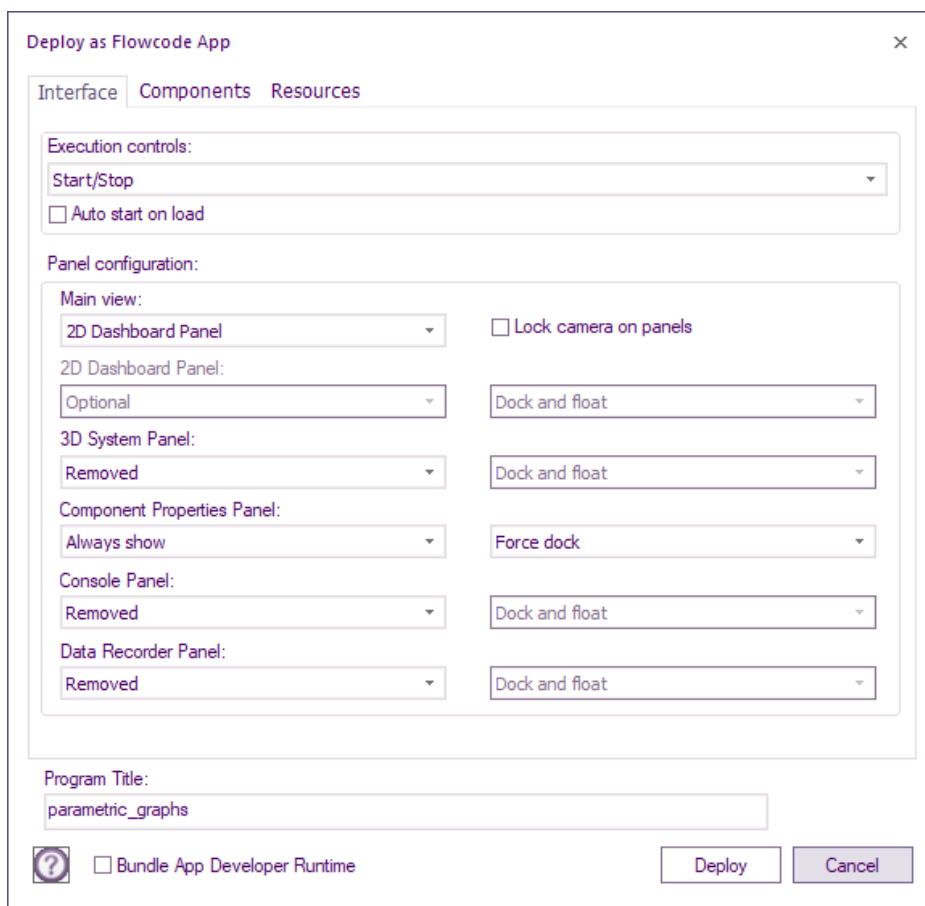


Рис. 6.6. Диалоговое окно экспорта

Обойдёмся для начала кнопкой *Deploy*. Появляется окно проводника, где предлагается указать место сохранения и имя файла. Пока оставим всё без изменений. Что завершается сообщением об удачном сохранении с указанием места расположения файла с расширением *.mscada*.

Теперь, запустив программу *Flowcode App Developer Runtime*, можно воспользоваться кнопкой *Open Project* меню выбора продолжения. Открывается файл с рабочей панелью Flowcode. Есть кнопка *Go*, которой и воспользуемся.

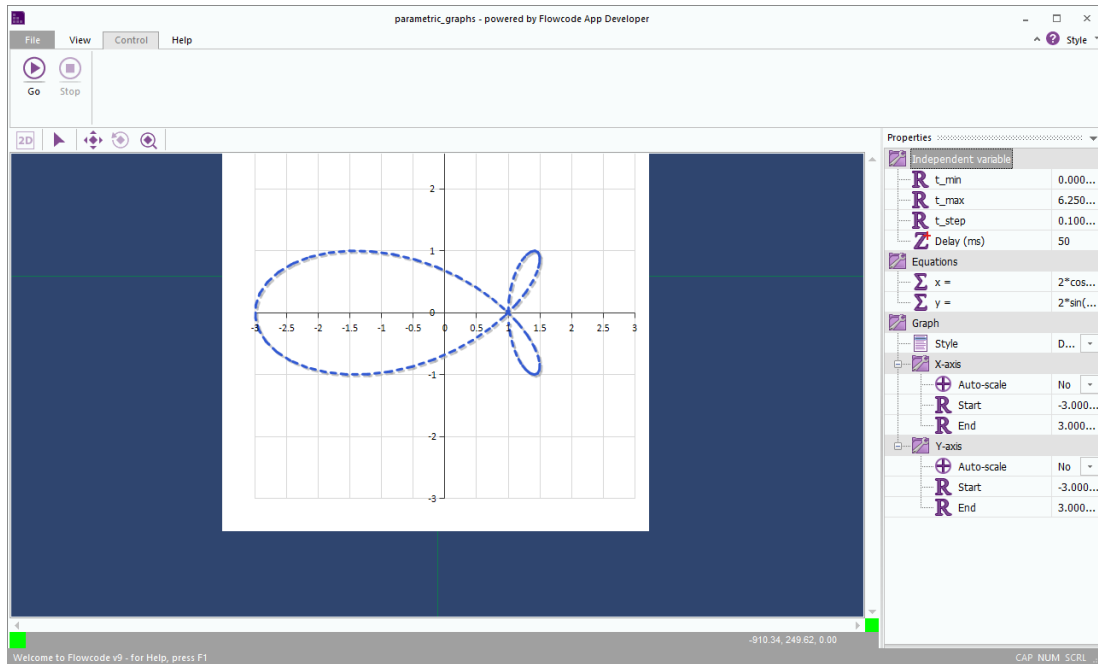


Рис. 6.7. Запуск открытого файла, полученного при экспорте

Теперь понятно, как можно запустить файл на рабочем компьютере. А как это сделать, например, на планшете? Чтобы это понять, я нахожу пример работы со SCADA, выполняю экспорт и получаю папку с несколькими файлами.

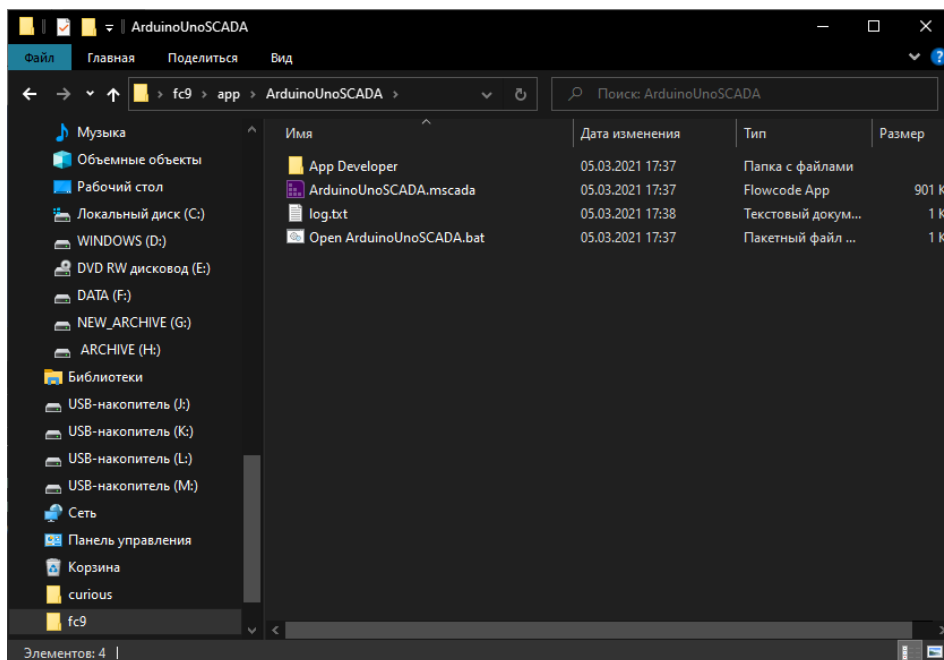


Рис. 6.8. Папка с файлами для продолжения работы с APP Developer

Можно запустить bat-файл, но я хочу поступить иначе. Я переименовываю файл, дав ему расширение .txt. Такой файл открывается в блокноте. И имеет простое содержание.

```
cls

set NetPath=%~dp0

@SET PF="%NetPath%\ArduinoUnoSCADA.mscada"

@start "" "%NetPath%\App Developer\Flowcode Runtime.exe" %PF%
```

Теперь поступим обычным образом – переименуем полученный ранее файл для графики в graph.mscada. Исправим bat-файл, изменив имя файла, который он вызывает. Сохраняем файл (не забываем использовать выбор «Все файлы») с расширением .bat. Теперь я создаю папку с именем graph, копирую в неё всё, включая экспортированный файл graph и новый bat-файл. Всё лишнее можно, наверное, удалить. Эту папку я переношу на свой планшет, где запускаю bat-файл graph.bat. В окне программы кнопка Go даёт продолжение (или завершение) этой части рассказа.

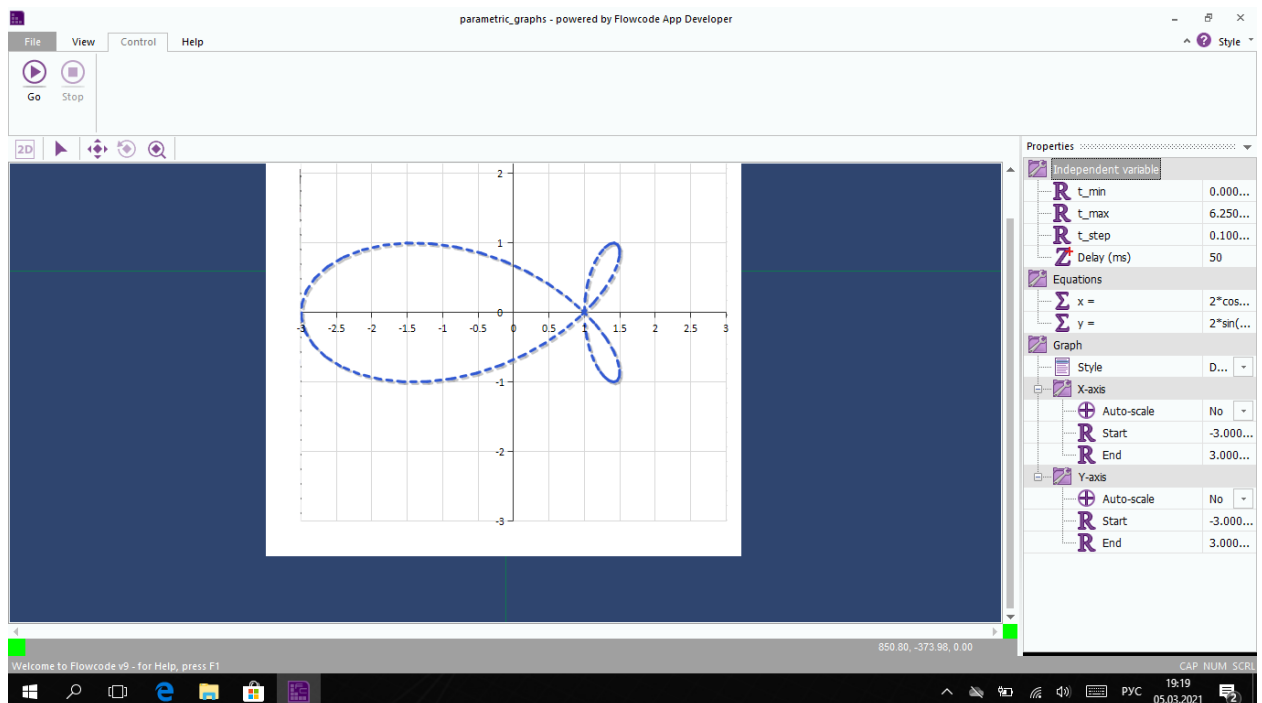


Рис. 6.9. Запуск программы на планшете

На планшете, признаюсь, пришлось использовать раздел настроек, где потребовалось переключить графическое ускорение с аппаратного (которого нет) на программное. Первые несколько раз я не догадался это сделать, программа отказывалась работать без объяснения причин, хотя сама программа *App Developer* работала.

Мне осталось понять, как самому повторить эти результаты. Намёк есть. Создадим пустой проект *App Developer*, в библиотеке компонентов (*Components Library*) обратим внимание на подраздел *Charts*, где выберем *XY Chart*, что и перенесём на рабочую панель. А в рабочее окно программы перенесем элемент *Simulation Macro* (так начинается проект, показанный ранее). В этом месте я вспоминаю, что до экспорта в подсказке рекомендовали отметить в свойствах...

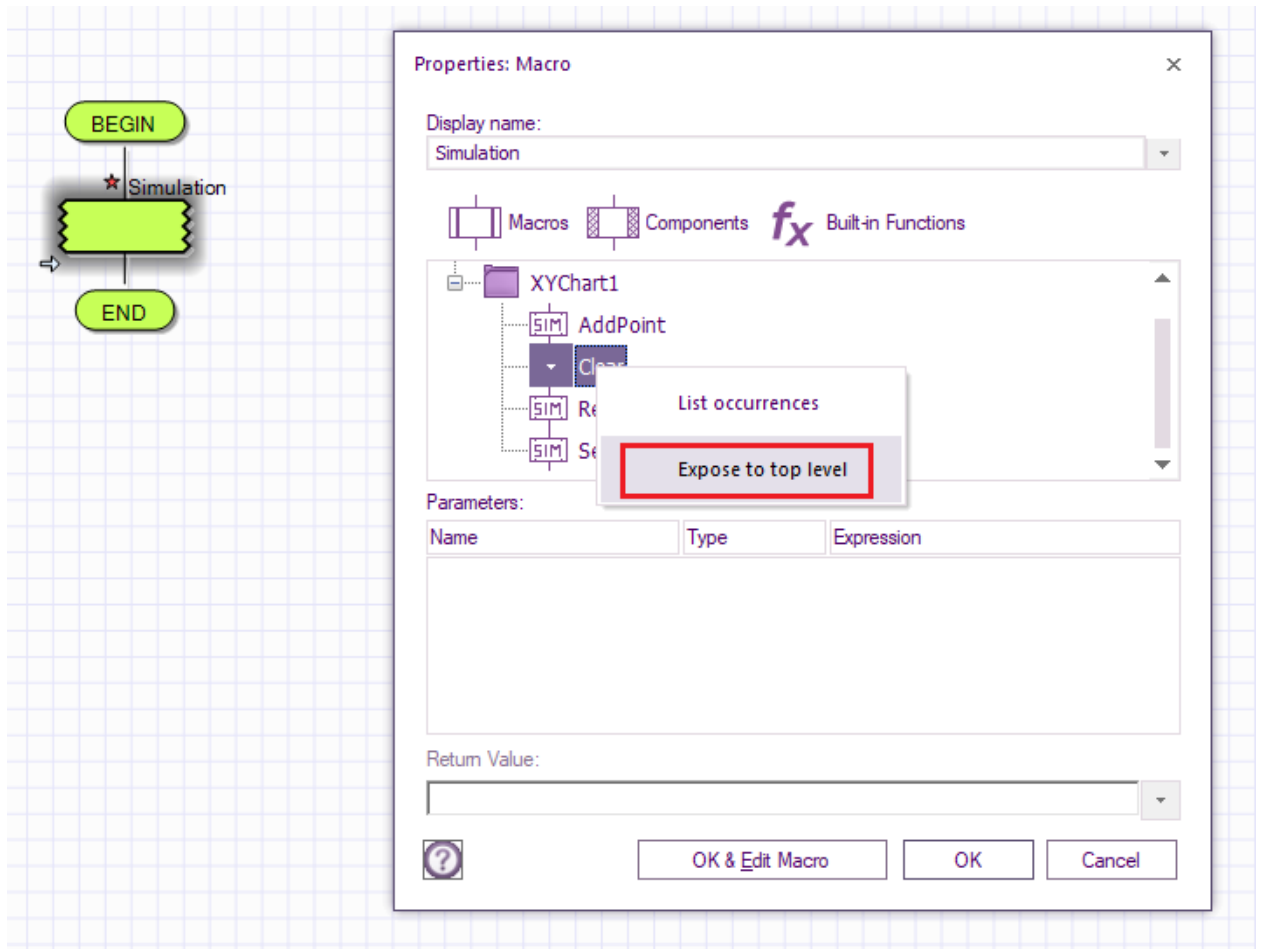


Рис. 6.10. Свойство для подготовки к экспорту

Выбрать это в свойствах предлагалось перед экспортом для всех нужных элементов. Я искал в свойствах разных элементов, но не нашёл. Только сейчас на него наткнулся. Бывает. Отсутствие этого действия можно продемонстрировать. Положим (можно же это предположить), я знаю, как собрать небольшую программу: в приложении для ПК будет кнопка, к ПК подключен через COM-порт модуль Arduino, на котором «крутится» программа. В приложении для ПК будет индикатор. Когда кнопка в приложении будет нажата, к модулю Arduino отправится команда, которая включит светодиод (его состояние контролирует индикатор в приложении).

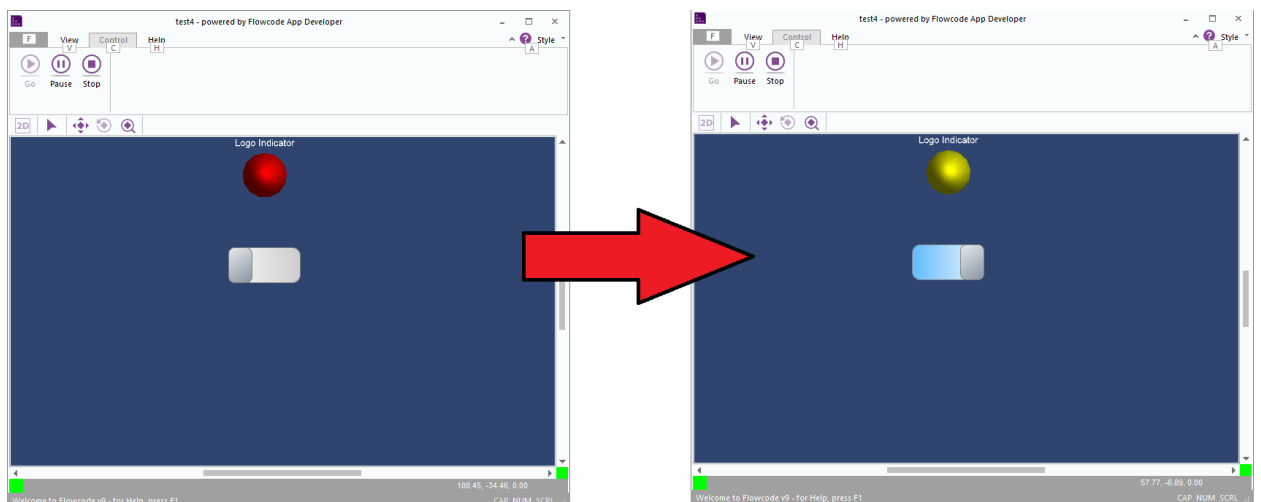


Рис. 6.11. Созданное приложение и его работа

Если не выполнить рекомендованную процедуру, не отмечать в свойствах *Expose to top level*, то созданная приложением исполнительная панель оказывается пустой. И добраться до элементов управления можно только добавив через операцию Вид (View) рабочую панель.

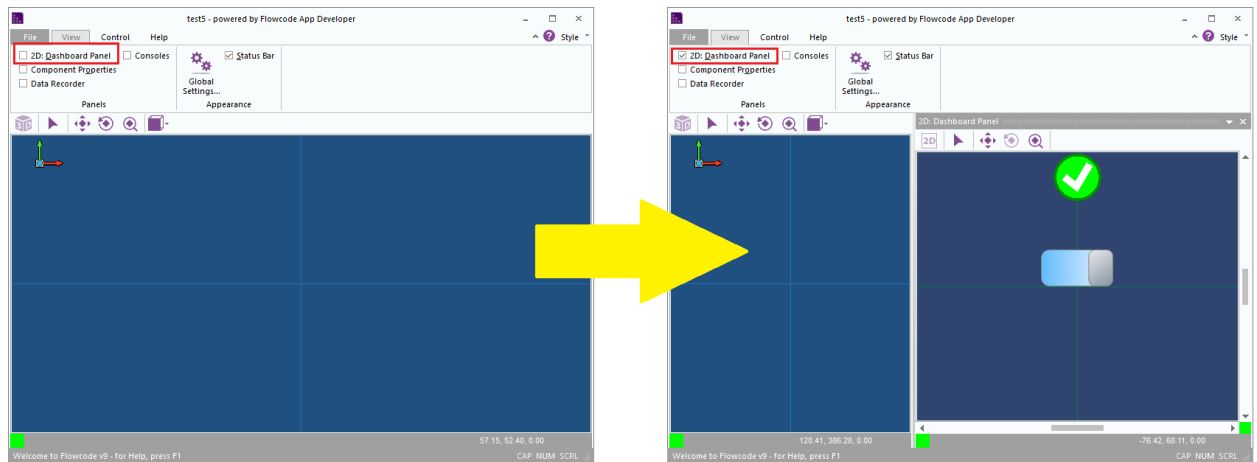


Рис. 6.12. Вид созданного приложения без предварительной отметки компонентов

Было бы, наверное, интересно попробовать программу с реальными модулями. Но завершить эту историю я хочу любимой цитатой: «Будь попрочнее старый таз...».