# Инструкция по установке шаблона программы пользователя AXCodeTemplate c GitHub

Чтобы получить актуальный шаблон пользовательской программы можно воспользоваться сервисом <u>github.com</u>, работающим на основе системы контроля версий Git.

- 1. Создайте аккаунт в системе контроля версий github.com.
- 2. Установите Git for Windows <u>https://git-scm.com/download/win</u>

i	При установке оставлять предлагаемые по умолчанию опции, но убедиться, что используются следующие: 1. "Git from the command line and also from 3rd-party software" на странице "Adjusting your PATH environment" 2. "Checkout Windows-style, commit Unix-style line endings" на странице "Configuring the line ending conversions" 3. "Default (fast-forward or merge)" на странице "Choose the default behavior of 'git pull" 4. "Git Credential Manager" на странице "Choose a credential helper"
---	--

 Создайте вилку репозитория AXCodeTemplate. Для этого перейдите на веб-страницу шаблона пользовательской программы на GitHub и в выпадающем списке "Fork" выберите "Create a new fork". В появившемся окне укажите название вилки и описание.



Torked from ELHARTLTD/AXCodeTemplate	S Pin 💿 Watch	0
រឹង main 🔺 ្រំ 1 Branch 📀 0 Tags	Q Go to file t Add file - Code -	J
This branch is up to date with ELHARTLTD/AXCodeTemplate:main .	រ៉ោ Contribute      ើG Sync fork   •	
😨 ELHARTLTD Шаблон программы пользователя v1.1.1	43bb247 · yesterday 🕚 1 Commit	

Вилка, созданная из репозитория AXCodeTemplate.

### 4. Копирование вилки на локальный компьютер.

Над списком файлов нажмите на кнопку "*Code*" и в выпавшем окне скопируйте ссылку на репозиторий.





Для того, чтобы удобнее работать с Git можно установить графический клиент TortoiseGit (<u>https://tortoisegit.org/</u>), либо предпочтительный аналог. Дальнейшие действия будут описаны, используя **TortoiseGit** и **Code-OSS.** 

На ПК откройте папку, в которой будет храниться репозиторий:

- 4.1. ПКМ в папке Git Clone;
- 4.2. В поле URL должна быть указана ссылка на репозиторий из предыдущего шага;
- В поле Directory указана папка для копирования репозитория (будет вложенной относительно текущей папки, может быть скорректирована);
- 4.4. Прочие поля остаются без изменений;
- 4.5. По нажатию на ОК происходит копирование репозитория из GitHub. Копирование прошло успешно если в плавающем окне появилась синяя надпись "Success".



После копирования репозитория с шаблоном программы пользователя, в нём можно начинать работу.

### 5. Создание коммита (сохранение изменений)

После внесения изменений в файлы скопированного репозитория, их можно сохранить, сделав коммит.

- Code-OSS:
  - 1. В панели Source Control изменения могут отображаться в двух вкладках:
    - 1. "Staged Changes" для изменений, которые будут добавлены в следующий коммит;
    - 2. "Changes" для изменений, которые не будут добавлены в коммит;



2. Файлы в "Changes" можно открывать и выбирать строки кода, которые будут добавлены в коммит (выделить строки - ПКМ -Stage Changes);

_			
		Go to Definition	F12
	27 void POU1()		
	28 {	Go to Declaration	
	<pre>29 static bool init = false;</pre>	Go to Type Definition	
	30 if (!init)	Go to Poferoncer	chi#±⊑12
		GO to References	3111117712
	<pre>32 init = true;</pre>	Peek	>
		Find All References	Shift+Alt+F12
		Show Call Hierarchy	Shift+Alt+H
	34+ 💡 print_debug("IP Address: %d.%d.%d.%		
	<pre>35+ eth1.getSettings().ipAddr[0],</pre>	Rename Symbol	F2
	<pre>36+ eth1.getSettings().ipAddr[1],</pre>	Change All Occurrences	Ctrl+E2
	<pre>37+ eth1.getSettings().ipAddr[2],</pre>	change All Occurrences	Cultr2
	<pre>38+ eth1.getSettings().ipAddr[3]);</pre>	Format Document	Shift+Alt+F
		Format Document With	
		Format Solaction	Ctd+V Ctd+E
	— // Код, выполняемый циклически:	Format Selection	
	<pre>print_debug("Hello!\n\r");</pre>	Refactor	Ctrl+Shift+R
	41+		
	42+	Share	>
	44 }	Stage Selected Ranges	Ctrl+K Ctrl+Alt+S

- 3. Напротив имени файла можно нажать "+" (Stage Changes) чтобы добавить файл в коммит целиком;
- B "Staged Changes" напротив имени файла можно нажать "-" (Unstage Changes), чтобы вернуть файл во вкладку "Changes" (также со строками в файле);
- 5. Для отмены изменений и возврата к состоянию файла в текущем коммите используется кнопка "Discard Changes" (иконка с разворачивающейся стрелкой);
- 6. В поле "Message" описываются изменения в коммите;
- 7. Коммит создан по нажатию галочки (Commit).
- TortoiseGit:
  - 1. ПКМ в папке с репозиторием Commit

предоставить доступ к	
🐴 Git Sync	
🕐 Git Commit	
TortoiseGit	>
Создать	>
Свойства	

- 2. В поле Message вводится описание изменений в коммите
  - 1. В окне changes made ставится галочка напротив файлов для добавления в коммит:

- Versioned files изменения в уже существующих файлах;
- Not versioned files Файлы отсутствуют в репозитории, нужно нажать ПКМ - Add;

Changes made (F5: re Check: <u>A</u> ll <u>N</u> one	fresh, double- Unversioned	click on file f	for diff): I Added Dele	ted <b>Modified Fi</b>	l <b>es</b> Submodules
Path	Extension	Status	Lines added	Lines removed	
🗹 🔄 src/pou1.cpp	.срр	Modified	1	1	
□ Staging support (EXPERIMENTAL) 1 files selected, 1 files total   □ Show Unversioned Files View Patch>>   □ Do not autoselect submodules View Patch>>					
Show <u>W</u> hole Proje	ect		Com	mit 🔻	Cancel Help

2. По нажатию "Commit" справа снизу создается коммит. Во всплывающем окне должна появиться синяя надпись "Success".

## 6. Отправка изменений.

После создания коммитов, их можно отправить в удалённый репозиторий.

• Code-OSS:

В панели Source Control справа сверху выбирается "..." - Push;

• TortoiseGit:

ПКМ в папке с репозиторием - TortoiseGit - Push - OK.

### 7. Создание ветки

- Code-OSS:
  - 1. В панели Source Control справа сверху выбирается "..." Branch:
    - "Create Branch" для создания ветки на основе текущей ветки/коммита;
    - "Create Branch From" для выбора ветки, на основе которой будет создана новая;
  - 2. Вводится имя новой ветки, Enter;
  - 3. VSCode автоматически переключится на новую ветку, если создано успешно.
- TortoiseGit:

- 1. ПКМ в папке с репозиторием TortoiseGit Create Branch;
- 2. В поле "Branch" вводится имя новой ветки;
- В секции "Ваѕе Оп" выбирается на основе какой ветки/коммита будет создана новая ветка;
- 4. С галочкой "Switch to new branch" git будет переключен на новую ветку;
- 5. Нажимается "ОК" для создания ветки.

# Приватный шаблон программы пользователя

- Для создания приватной копии шаблона программы пользователя необходимо создание нового приватного репозитория и клонирование оригинального репозитория, поскольку GitHub не позволяет напрямую создавать приватные вилки.
- 2. Чтобы предоставить другим пользователям доступ нужно перейти в настройки репозитория, во вкладке Access, в разделе Collaborators, в окне Manage Access выбрать пользователей, которым будет предоставлен доступ.

	Access		
-	A Collaborators		
Manage access			
	Ø		
You haven't invited any collaborators yet			
	Add people		