

# Glitch

A Visual Compiler

2025.04.26

## Table of Contents

Introduction.....	3
Arduino Special Functions.....	4
Creating New Diagrams.....	5
Editing Diagrams.....	6
Flow-Control Objects.....	7
Function Objects.....	8
Operating Systems.....	9
SQL Injections.....	10
Wiring Objects.....	11

## Introduction

Glitch is a visual compiler. The software interprets block diagrams (blueprints) and translates the diagrams into Arduino intermediate C source. Glitch is extensible with other frameworks, for example, the C programming language.

Glitch is functional on any operating system where Qt LTS is supported. Qt 5.5.1 is supported for PowerPC and other operating systems. Qt 4.8.x is considered obsolete and is not supported.

The source of Glitch is available at <https://github.com/textbrowser/glitch>.

## Arduino Special Functions

The Arduino programming interface requires two special functions, `loop()` and `setup()`. The functions are automatically assigned to an Arduino diagram after a diagram is created. The objects representing the two methods are assigned static geometries and are situated at the top-left corner of a diagram.

## Creating New Diagrams

New diagrams may be created via File → New Diagram → Arduino. After a diagram is initialized, editing may begin. To create an object, drag-and-drop it from the Categories tree widget. Objects may also be added from the copy buffer via paste events. Diagram information is recorded in a portable SQLite database. Each object type offers general and specific properties which are accessible via context menus and property editors.

## Editing Diagrams

Existing objects may be edited via direct interactions. A context menu is also available for each object. The context menu contains generic and specific properties. Copying and pasting objects are also allowed. A single redo / undo stack provides rich redo / undo behavior.

## Flow-Control Objects

Objects such as switch statements may contain Arduino instructions. If a flow-control object contains at least one instruction, the flow-control object's name will be underlined.

Glitch

## Function Objects

A Glitch function having an underlined name indicates that the function has a non-empty body. That is, the function contains at least one Arduino instruction.



## Operating Systems

Glitch supports Android, FreeBSD, Linux, MacOS, OS/2, OpenBSD, and Windows. Generally, the application should be compatible with any operating system where a modern Qt is supported. The software has also been tested on a variety of architectures, including AMD, ARM, Apple Silicon, PowerPC, and SPARC.

Glitch

## SQL Injections

All Glitch SQL queries are parameterized. Prepared SQL statements are resilient against SQL injections.

## Wiring Objects

Wired objects designate a graphical relationship between the wired objects. For example, a variable object wired to a function object suggests one of two things. For a main-diagram function, a wired variable connected to the function implies that the function has one parameter. For a non-main-diagram function, a wired variable (or another object type) suggests that the function be issued with the wired input. To wire two objects, click on either the input or output side of one of the objects and then click on the input or output side of the second object. Some objects lack input and output ports. If a wiring indicator is not shown for an object on a hover event over an input port, the object is maximally-wired on the input port.

## Index

AMD.....	9	OpenBSD.....	9
Android.....	9	OS/2.....	9
Arduino.....	3	f. pasting.....	6
ARM.....	9	PowerPC.....	3, 9
block diagrams.....	3	Qt.....	9
blueprints.....	3	Qt 5.5.1.....	3
Categories.....	5	Qt LTS.....	3
context menu.....	6	redo / undo stack.....	6
Copying.....	6	setup().....	4
FreeBSD.....	9	SPARC.....	9
Glitch.....	3	SQL.....	10
Linux.....	9	visual compiler.....	3
loop().....	4	Windows.....	9
MacOS.....	9	Wired.....	11