

Glitch

A Visual Compiler

2025.12.30

Table of Contents

Introduction.....	3
ASH.....	4
Arduino Special Functions.....	5
Creating New Diagrams.....	6
Editing Diagrams.....	7
Flow-Control Objects.....	8
Function Objects.....	9
Home Path.....	10
Object Order.....	11
Operating Systems.....	12
SQL Injections.....	13
Wiring Objects.....	14

Introduction

Glitch is a visual compiler. The software interprets block diagrams, or 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>.

Glitch

ASH

ASH (a shell) is an interactive interface included with a diagram. It is usually situated at the bottom of a design window. Within ASH, shortcut instructions may be executed. For example, the instruction **save** writes a diagram's contents to its respective database file. The instruction **help** lists the available commands.

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 assign an object to a diagram, drag-and-drop the object from the Categories tree widget. Objects may also be added from the copy buffer via paste events. Each object type offers generic and specific properties which are accessible via context menus and property editors.

Diagram information is recorded in a portable SQLite database.

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 diagram includes a single redo / undo stack which 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 object, 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 object.

Glitch

Home Path

The environment variable `GLITCH_HOME` describes the directory where Glitch content will be stored for the current account. If `GLITCH_HOME` is not configured, the following logic describes the home path of Glitch for the current account.

Unix

The account's home path + `/ + .glitch`. For example, `/home/saturn/.glitch`.

Windows

The application's current directory + `\ + .glitch`. For example, `C:\Glitch\.glitch`.

Object Order

An object is assigned an order in the parent diagram. The order is associated with the object's placement in the generated source file. Objects of lower order are placed at the beginning of the generated file. The special Arduino methods **loop()** and **setup()** are placed at the end of the generated source file.

Glitch

Operating Systems

Glitch supports Android, FreeBSD, Linux, MacOS, OS/2, OpenBSD, and Windows. Generally, the application is 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 the first object 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. Version 2025.05.10 introduced an input-full Port Colors property.

Index

AMD.....	12	OS/2.....	12
Android.....	12	pasting.....	7
Arduino.....	3, 5	PowerPC.....	3, 12
ARM.....	12	Qt.....	12
block diagrams.....	3	Qt 5.5.1.....	3
Categories.....	6	Qt LTS.....	3
context menu.....	7	redo / undo stack.....	7
Copying.....	7	setup().....	5
FreeBSD.....	12	SPARC.....	12
Glitch.....	3	SQL.....	13
Linux.....	12	visual compiler.....	3
loop().....	5	Windows.....	12
MacOS.....	12	Wired.....	14
OpenBSD.....	12		