

VISUINO GUIDE FOR BEGINNERS



Start your journey today and transform your ideas into reality with Visuino

MODULE 1. Introduction: Visuino Guide for Beginners

MODULE 2. What Kind of Equipment Does Visuino Require?

MODULE 3. Installing Visuino

MODULE 4. <u>Navigating the Visuino Interface</u> <u>4.1 Main Menu Bar</u> <u>4.2 Toolbox Bar</u>

MODULE 5. <u>Project Workspace</u> 5.1.1 Selecting the Microcontroller Board 5.1.2 Working with Microcontroller Board Shields

MODULE 6. Components Toolbar

6.1 The Components Categories 6.2 The Components Window Options

MODULE 7. Component Properties Window

MODULE 8. Preview and Navigation

MODULE 9. Visuino Panel

9.1.1 Help Tab 9.1.2 Build Tab 9.1.3.1 Terminal Tab 9.1.4 Errors Tab 9.1.5 Platforms Tab 9.1.6 Libraries Tab

MODULE 10. Advanced Features in the Professional Version

<u>10.1 Instruments Tab</u> <u>10.2 Professional Version</u>

MODULE 11. Master analog values, control structures, and switch logic in Visuino

MODULE 12. Poject Examples

11.1. LED Blink11.2. Weather Station using Visuino Instruments11.3 How to Use I2C OLED Display11.4 Ultrasonic Range Finder11.5 How to Save Sensor Data Temp & Time to SD Card11.6 Ethernet Shield & MAX7219 LED Display - Get Time & Date From theInternet

MODULE 13. Visuino Resources

Introduction: Visuino Guide for Beginners

Welcome to the exciting world of Visuino, where the power of visual programming meets the limitless possibilities of microcontrollers and electronics. If you've ever wanted to create your own interactive devices, automate your home, or explore the realms of IoT (Internet of Things), you're in for an incredible journey.

Visuino Guide: A Comprehensive Guide to Visual Programming is your key to demystifying the complex world of Arduino and microcontroller programming. Whether you're a seasoned electronics enthusiast or an absolute beginner, this book is designed to be your trusted companion, guiding you step by step into the realm of visual coding.

In a world where technology and innovation are advancing at an astonishing pace, Visuino empowers you to harness the potential of hardware without the need for intricate lines of code. By using intuitive, drag-and-drop blocks, you'll be able to control sensors, displays, motors, and other electronic components with ease, allowing you to bring your ideas to life quickly and efficiently.

Throughout the pages of this book, you will embark on an exciting learning journey. We'll cover the fundamentals of Visuino, gradually building your skills and confidence. You'll discover how to create Arduino-based projects, experiment with sensors, and even delve into advanced concepts like wireless communication and data logging.

Our goal is to equip you with the knowledge and skills to turn your innovative ideas into functional, real-world projects. From home automation to robotics, from interactive art installations to smart gadgets, the possibilities are endless, and Visuino is your gateway to making them a reality.

So, whether you're a hobbyist, a student, a DIY enthusiast, or a professional looking to simplify your workflow, let's embark on this exciting journey together. By the end of this book, you'll be a Visuino wizard, ready to design and program your creations effortlessly.

So, let's dive in, and let the adventure begin!

What Kind of Equipment Does Visuino Require?

Before we dive into the exciting world of Visuino and visual programming, it's essential to understand the equipment you'll need to get started. The beauty of Visuino lies in its ability to bridge the gap between software and hardware, enabling you to bring your electronic creations to life with ease. Let's explore the essential equipment and tools that will be your companions on this journey.

2.1. Hardware Components

2.1.1. Microcontroller Board

A microcontroller board is the heart of your projects. Visuino is highly compatible with various Arduino boards, including the Arduino Uno, Arduino Mega, and many others. These boards provide the processing power needed to execute your programmed instructions.

2.1.2. Sensors and Modules (Optional)

Depending on your project's goals, you'll require sensors or modules (e.g., temperature sensors, motion sensors, light sensors, servos, motors, LEDs) to interact with the physical world. The choice of these components will depend on the specific functions you want your project to perform.

2.2. Computer and Software

2.2.1. Computer

You'll need a computer to run Visuino. Visuino is a Windows-based application, so a Windows PC or laptop is recommended. However, it is also possible to run Visuino on a virtual machine if you're using a different operating system.

2.2.2. Visuino Software

The core of this book is the Visuino software itself. You can download Visuino from the official website (<u>www.visuino.com</u>). Notably, there are two versions of Visuino available:

- **Standard Version:** The standard version of Visuino provides you with a wide array of features for creating and programming your electronic projects. It's an excellent choice for hobbyists and enthusiasts looking to explore the world of visual programming.
- **Professional Version:** The professional version offers advanced capabilities and additional components for more complex projects. It's ideal for those seeking to take their skills to the next level and tackle intricate electronic designs.

In the following chapters, we'll explore the intricacies of microcontroller boards, sensors, modules, and more. So, gather your equipment, set up your workspace, and let's begin this exciting journey into the world of visual programming with Visuino.

2.3. Additional Tools

2.3.1. USB Cable

To connect your microcontroller to your computer, you'll need a USB cable. The type of USB cable required depends on the specific microcontroller board you are using.

2.3.2. Breadboard and Jumper Wires

For prototyping and experimenting with electronic circuits, a breadboard and jumper wires are invaluable tools. These allow you to build and test your circuit connections without soldering.

Now that you have an overview of the equipment and tools you'll need, let's delve deeper into each component and learn how to set up your workspace for a seamless Visuino experience. In the following chapters, we'll explore the intricacies of microcontroller boards, sensors, modules, and more. So, gather your equipment, set up your workspace, and let's begin this exciting journey into the world of visual programming with Visuino.

Installing Visuino

Now that you've gathered your equipment and are ready to get started, the next step is to install Visuino on your computer. Here's a step-by-step guide to help you with the installation process:

Step 1: Download Visuino

- 1. Visit the official Visuino website at <u>www.visuino.com</u> or <u>http://www.visuino.eu</u>
- 2. Navigate to the "Download" section.
- 3. Choose the version of Visuino you'd like to install: the Standard or Professional version. Make sure to select the one that best suits your project needs.
- 4. Click on the download link for your chosen version.

Step 2: Run the Installer

- 1. Locate the downloaded Visuino installer file on your computer (usually in your "Downloads" folder).
- 2. Double-click the installer file to run it. You may be prompted to provide administrator permissions.
- 3. Follow the on-screen instructions to complete the installation process. You can typically accept the default installation settings.

Step 3: Launch Visuino

Once the installation is complete, you can launch Visuino by either clicking on its desktop icon or finding it in your list of installed programs.

Step 4: Verify Installation

- 1. After launching Visuino, take a moment to explore the interface. Familiarize yourself with the various components, such as the toolbox, workspace, and properties panels.
- 2. To ensure that Visuino is working correctly, you can start a new project, open an existing one, or experiment with one of the example projects provided with the software.

With Visuino successfully installed, you're now ready to embark on your journey into the world of visual programming. In the upcoming chapters, we'll guide you through the software's features, functions, and capabilities. You'll soon be creating your own projects and bringing your electronic ideas to life with ease.

Let's begin by getting acquainted with the Visuino interface and understanding how to use its features effectively.



Navigating the Visuino Interface

Visuino boasts a user-friendly and intuitive interface designed to simplify the process of creating electronic projects through visual programming. As you embark on your journey with Visuino, it's essential to become familiar with the primary elements of the software's interface. Let's explore these elements to ensure you can navigate Visuino with ease:

- 1. Main Menu Bar & Toolbox Bar
- 2. Project Workspace
- 3. Components Toolbar
- 4. Component Properties Window
- 5. <u>Preview and Navigation</u>
- 6. <u>Visuino Panel</u>

The menus provide access to a range of functions that are vital for creating and managing your electronic projects. Let's explore these menu options to ensure you can make the most of Visuino's capabilities:

4.1 Main Menu Bar

File Project Edit Search View Arduino Help

The main menu bar is located at the top of the Visuino window and provides access to various functions and features. Here's an overview of what you'll find in the menu bar:

4.1.1. File Menu



The File menu offers a set of functions that enable you to manage your projects:

- New: Create a new project.
- **Open:** Open an existing project.
- Open Demo: Access various demo projects for exploration and inspiration.
- Save: Save your project.
- Save As: Save your project with a different name or in a different location.
- Save Picture: Save the project main window as an image.
- Reopen: Open recently opened project.
- **Print:** Print your project.
- **Print Setup:** Configure the print settings.
- Exit: Exit Visuino.

4.1.2. Project Menu (Professional Version Only)



The Project menu in the Professional version provides tools for managing sub-diagrams:

- Add Sub Diagram: Add a sub-diagram to your project.
- Delete Sub Diagram: Remove a sub-diagram.
- Export Sub Diagram: Export a sub-diagram to a file.
- Import Subdiagram from a File: Import a sub-diagram from an external file.

4.1.3. Edit Menu



The Edit menu helps you manage your project components and their properties:

- Undo: Reverse the previous action.
- **Redo:** Restore the previously undone action.
- Cut: Remove and copy selected components to the clipboard.
- **Copy:** Copy selected components to the clipboard.
- Paste: Insert copied or cut components from the clipboard.
- Copy as Picture: Copy the project view as an image to the clipboard.
- Arrange Filters: Automatically arrange components for an organized project layout.
- Auto Route: Enable or disable automatic connection routing.
- Preferences: Enable Version number or Project name in the title, Set Auto save.

4.1.4. Search Menu



The Search menu allows you to search through project elements, properties, and components:

• Find: Open a search window at the bottom with options to search through types, elements,

and properties of boards or components.

- Find Next: Navigate to the next search result.
- Find Previous: Move to the previous search result.

4.1.5. View Menu



The View menu offers settings to customize the appearance of the main window:

- Grid: Turn the grid on or off in the main window.
- **Rulers:** Enable or disable rulers in the main window.
- Dark Theme: Switch between the dark theme and the default theme.

4.1.6. Arduino Menu



The Arduino menu provides functions related to the code generation and uploading process:

- **Build:** Compile the code.
- Send to Arduino IDE: Launch the Arduino IDE and open the code in it.
- Upload: Upload the code to the microcontroller board.
- Refresh Arduino Code: Update the code based on the latest library or file changes.
- Configure: Access a window to set or change various Arduino-related configurations.

4.1.7. Help Menu



The Help menu contains options for assistance and information:

- Enter Registration Key: Input a Visuino registration key.
- Context Help: Open a wiki webpage displaying help topics related to selected components.
- About: Access information about Visuino.

Understanding the functions available in these menus is important as they provide you with powerful tools for creating and managing your electronic projects with Visuino. In the upcoming sections, we will explore these menu options in more detail, allowing you to harness Visuino's full potential.

4.2 Toolbox

File Project Edit Search View Ardui	o Help	
🐒 🗫 🕞 🔚 🔛 📑 🏢	🕤 - (~ - Zoom: 100% - 🗸 🔍 🔍 🧱 🌠 📉 🕅 🧕 象 🧟 🥌 🌜	
	🥖 Main 🔝 Code	
		i.

Navigating the Visuino Toolbox

As you familiarize yourself with the Visuino interface, it's important to understand not only the menu options but also the functions available through the **Toolbox**. The toolbox provides you with quick access to important actions and tools to streamline your project development. Let's explore these toolbox buttons to ensure you can make the most of the Visuino's capabilities:

4.2.1 Toolbox Buttons

The toolbox is equipped with a variety of buttons that offer quick access to essential actions and tools:

- New Project: Create a new project.
- Open Demo: Access a range of demo projects for exploration and inspiration.
- **Open File:** Open an existing project.
- **Open File (Arrow Down Submenu):** Open recent projects, providing easy access to your recent work.
- Save: Save your project.
- Add Subdiagram (Professional Version): Add a subdiagram to your project, enhancing its organization.
- **Delete Subdiagram (**Professional **Version):** Open a window to select a subdiagram for deletion.
- Show/Hide Rulers: Toggle the display of rulers in the main window.
- Show/Hide Grid: Toggle the grid on or off in the main window.
- Undo: Reverse the previous action.
- **Redo:** Restore the previously undone action.
- Zoom: Drop down box for selecting Zoom percentage.
- Zoom In: Zoom in on the project view for a closer look.
- Zoom Out: Zoom out to see a wider perspective.
- Zoom Off: Turn off the Zoom.
- Auto Arrange: Automatically arrange components for an organized project layout.
- Enable/Disable Auto Routing: Control automatic connection routing.
- Delete Selection: Remove selected components from your project.

- Find (Search Window Opens): Open a search window at the bottom with options to search through types, elements, and properties of boards or components.
- Compile/Build: Compile and build your project.
- **Open Build Folder:** Open the location where the project file was generated.
- Upload to Arduino: Upload your project to the microcontroller board.
- **Refresh the Generated Arduino Code:** Update the code based on the latest library or file changes.
- Send to Arduino IDE for Compilation: Launch the Arduino IDE and load the code.
- **On/Off Dark Theme:** Switch between the dark theme and the default theme for the main window and preview window.

Understanding these toolbox buttons is very important, as they provide you with quick access to essential functions and actions, making your work in Visuino more efficient and productive. In the upcoming sections, we will explore how to use these toolbox buttons effectively to create and manage your electronic projects.

Project Workspace

The central area of the Visuino interface is your project workspace. This is where you'll arrange and connect components, creating the logic and flow of your project. You can think of this area as your digital workbench, where your circuits take shape.



5.1. Selecting the Microcontroller Board or Shield

5.1.1 Selecting the Microcontroller Board

At the heart of the Project Workspace is the space where you select and configure the microcontroller board or shield that will be the core of your project. This selection is a fundamental step in your project setup, as it determines the hardware that your project will run on. Whether you're working with an Arduino Uno, Arduino Mega, or other supported boards, or if you're utilizing specific shields for added functionality. Once your selection is made, you can start adding components and defining connections to create your project's functionality.

This section serves as the foundation of your project, so it's essential to choose the appropriate

board or shield that aligns with your project goals. The Visuino interface simplifies this process, ensuring that you can quickly set up the hardware side of your project and focus on the creative aspects of design and programming.

To select the **Microcontroller** board you simply click on the "**Tools**" button on the Arduino component (Picture 1) in Visuino When the dialog appears, select the **Microcontroller board** for example "**Arduino UNO**" as shown on the Picture.

5.2 Working with Shields

In Visuino, the ability to incorporate shields into your electronic projects is a valuable feature. Shields, which are add-on boards, can expand your project's capabilities and add new functionalities. Let's explore how to add and configure shields in Visuino:

5.2.1. Adding Shields

To add a shield to your project, follow these steps:

1			* * * * * * * * * * * * * * *		1	1				1	1	1			1
÷.			Arduino I Ino	2	4		-					2			ę
•	•	•		N	•	-		•	•	•			•		
			🦈 Serial	45	S	ele	ct	Boa	ard						÷
•			👘 Serial[0] (9600)		1	de		iel	de						
			¢⇔ln Sending 🦣		_	uu	1.51	ner	us.				_		÷
			Out	þ	÷	÷	\cdot	\cdot	•			÷			
1			I2C Channels							22					i.
1			ISC 12C												1
1			100					\sim		\sim	2				
1			¢i∰In Out∭	þ								22			è
•			Request Л	<u> </u>											Ċ,
				_							3	3		3	

- 1. Next to the "Tools" button, you'll notice a dropdown arrow menu.
- 2. Click the dropdown arrow menu, and you'll find the "Add Shields..." option. Please note that not all boards support adding shields.
- 3. Select "Add Shields..." to proceed.
- 4. A new window will open, displaying all the supported shields for the selected board. You'll find a range of shields, including joysticks, communication shields, displays, and more.

🍟 Arduino Uno	* →	n an an an an an an an an a	8 26 26 26 26 26 26 26		
🐡 Serial 🐡 Serial[0] (960 약 In)0) Sending 💽 🗗	n 24 28 28 28 28 28 28 2 n 24 28 28 28 28 28 28 2 n 24 28 28 28 28 28 28 28 2	6 16 16 16 17 18 18 6 18 16 18 17 18 18 6 18 18 18 18 18 18		
I2C Channe Kg I2C	🥌 Shields 守 ▾ 侯 ▾ 🔞 🙀		B		□ × _ 9 % 12
Sin Signal	Name Kadafruit V1.2 Shield	Type Adafruit V1.2 Shield	Funduino	Joystick Shield ation	
Digital			Action Served	iman/Xiro Robot Mo axim Motor Driver S dafruit V1.2 Shield Motors	tor Shield hield
Digital 2]			> - 💓 Stepp 🗱 Displays 	er Motors olor Touch Screen D uit LCD 5 Button Ke	Display ILI9341 Sh ypad I2C Shield
Digital O Digital (Clo Digital O Digital (5) (PWM0B Digital			Adafru DFRot > 🛣 Robots > 💇 General Vi	uit RGB Light LCD 5 bot LCD 5 Button Ke sion	Button Keypad I2 ypad Shield
Analog (PWM) Objectal[6] (PWMOA Digital Analog (PWM)			> Elegoo > 🍖 Analog ✓ 🔯 Digital	uit V1.2 Shield	
Digital O Digital [7] (Ena	ta)	n an an an an an an an	<		>

- 5. To add a shield, you have several options:
 - Click on the "Add New Item" ^{•••} button: Select the shield on the right side and click on the "Add New Item" button.
 - **Drag and Drop:** Select the shield on the right side of the window and drag it onto the left side.
 - **Double-Click:** Double-click on the shield you want to add.
 - **Right-Click:** Right-click on the shield and select "Add" from the popup menu.

5.2.2. Configuring Shield Properties

Once you've added a shield to your project, you can configure its properties in the Component Properties Window. Each shield may have specific settings and parameters that you can adjust to tailor its behavior to your project's needs. These properties allow you to fine-tune the operation of the shield, ensuring it integrates seamlessly with your project's logic.

By incorporating and configuring shields, you can extend the capabilities of your electronic projects in Visuino. In the chapters ahead, we'll dive deeper into working with specific shields and explore how to maximize their potential within your projects.

The Components Toolbar

The Components Toolbar in Visuino is where you access a wide range of building blocks to create your electronic projects. These components are organized into various categories to help you quickly find the elements you need. Let's explore the categories available in the Components Toolbar:



6.1 The Components Categories

- Converters: Components for converting data or signals from one format to another.
- Analog: Components related to analog input and output, ideal for working with continuous voltage values.
- Text: Components that enable the manipulation and display of text.
- Communication: Components for establishing communication and data exchange between

devices.

- Data Sources: Components that provide data or information sources for your projects.
- Data Sinks: Components that receive and process data from other sources.
- **Digital:** Components for handling digital input and output, suitable for binary signals.
- **Displays:** Components to visualize data, information, or graphics.
- Integer: Components designed for working with signed whole numbers.
- Measurement: Components for measuring physical quantities like distance or temperature.
- **Remote Controllers:** Components that facilitate remote control and communication with devices.
- **Output:** Components that produce an output, such as sound or light.
- Motors: Components for controlling and managing motors and motor-driven devices.
- Input/Output: Components that serve as interfaces for both input and output data.
- M5 Stack: Components specific to M5 Stack hardware, offering specialized functionalities.
- Seed Studio Groove System: Components tailored for the Seed Studio Groove System, providing integration with Groove modules.
- Filters: Components to filter and process data.
- Color: Components related to color analysis and processing.
- **Synchronization:** Components that enable synchronization and timing control in your projects.
- Math: Components for performing mathematical operations and calculations.
- **Complex:** Components for working with complex numbers and data.
- Audio: Components for handling audio signals and sound output.
- **Binary:** Components designed for manipulating binary data.
- Unsigned: Components for working with unsigned integer data types.
- Date/Time: Components related to date and time management.
- Gates: Components for logical operations and control.
- Memory: Components for memory management and data storage.
- Flip-Flops: Boolean Flip-Flop components.
- Generators: Components to generate signals and data.
- Image: Components for image processing and display.
- **3D:** Components related to three-dimensional data and graphics.
- 2D: Components for two-dimensional data and graphics.
- Quaternion: Components for working with quaternion data.
- Astronomy: Components for astronomy-related calculations and data.
- Touch: Components for handling touch input and interaction.
- **Control (In Pro version):** Components that offer advanced Industrial control and automation features in the Professional version of Visuino.

These categories serve as your toolbox for building electronic projects with Visuino.

6.2 The Components Toolbar Options

The Components Toolbar in Visuino offers a wide array of components to choose from when building your electronic projects. To help you efficiently navigate this extensive library, the Components Toolbar provides various features for filtering and managing components. Let's explore these functionalities:



6.2.1. Filter Box

The Filter Box is a powerful tool for searching and managing components. It includes the following options:

• EditWindow: Type part of the name of the component you are looking for.



• Suggest (The Wizard Hat): When you press the Suggest button, Visuino will suggest compatible components based on the component currently selected in the Project

Workspace. Suggestions are made considering pin types and other factors. There are three suggestion types: Best Match, Partial Match, and No Match. Partial match components are slightly grayed out, and No Match components are fully grayed out. You can also choose to hide No Match components. When Sort is enabled, the Best Match components are displayed at the top.

- Auto Hide Button : Enabling Auto Hide will automatically close other open categories when you open a category. This feature helps streamline your component selection process.
- Expand All^¹: The "Expand All" button expands all the categories, making it easier to browse and locate specific components.
- Collapse All[©]: The "Collapse All" button collapses all the categories, providing a more condensed view of the available components.
- Name As Button S: By clicking on the "Name As" button, you can set a custom name for components. First, type the custom name, then drag the component to the main window or click on it to add it with the custom name.



- Add Numbers Button¹²³: The "Add Numbers" button allows you to add numbers to the added components. If it's turned off, components will not be numbered if no other component has the same name.
- Toolbox Options Button $\overset{\checkmark}{\sim}$: Clicking the "Toolbox Options" button opens a toolbox options window. Here, you can configure settings such as auto-hiding hints after a certain interval in seconds, setting a large font for hints, and changing the component view to Treeview.





TreeView

The Treeview in Visuino is a versatile user interface element designed to display hierarchical data in a structured, easy-to-navigate format. This view is particularly valuable when organized categories and components or options into categories and subcategories, making it ideal for applications such as selecting organized components or navigating complex data structures.





Component Properties Window

The Component Properties Window is a crucial part of the Visuino interface. It provides detailed information and customization options for the selected components within your project. This window allows you to configure microcontroller board or component-specific settings, such as pin assignments, parameters, and behavior. Understanding how to use this window effectively is essential for tailoring the functionality of your project components.

7.1. Additional Features in the Component Properties Window

In addition to the fundamental functionalities, the Component Properties Window offers several powerful features that enhance your control and organization of component settings. Let's explore these features:

Filter and Search

At the top of the Component Properties Window, you'll find a Filter Edit Box and a Filter by Name button. These tools allow you to search for specific properties quickly, even in projects with numerous components. The Filter Edit Box is where you can input your search terms, and the Filter by Name button executes the search.

Expand and Collapse

For efficient property management, Visuino provides two buttons: Expand All and Collapse All. These buttons allow you to expand or collapse all the properties in the Component Properties Window with a single click, simplifying navigation and organization.

Categorization

The Categorize All button is a valuable tool for organizing properties. Clicking this button categorizes the properties, making it easier to locate and understand them. This feature is particularly helpful when dealing with complex components that have numerous settings.

Bindable Properties

Some components have properties that can be bound to other components or variables, creating dynamic behavior in your project. The Show Bindable All button helps you identify these bindable properties, streamlining the process of connecting and controlling components.

Toolbox Options Button Allows you to access the Font Settings option.

The Font Settings option allows you to increase the font size within the Component Properties Window, making it easier to read and work with properties. This feature can be particularly useful when you need to make the text more legible, especially on high-resolution screens.

Understanding and utilizing these additional features in the Component Properties Window will help you work more efficiently and maintain a well-organized workspace as you design and develop your electronic projects in Visuino.

7.2 Advanced Component Settings

In Visuino, certain components offer more advanced settings that go beyond the Component Properties Window. You'll notice that some components have a "…" button within the properties window that opens an additional window with a set of elements. These elements provide you with an even deeper level of customization. Moreover, these advanced settings may also reflect in the Component Properties Window as extra parameters, offering fine-grained control over your project's components. Understanding how to navigate and utilize these advanced settings is a valuable skill when crafting complex electronic projects.

										10 10	2.2
	무	👼 Elements		100.0	\Box \times		2		2	2 Q	
Properties		😝 + 😽 + 🛞 🙀			9 🌍 12	· ·		10 10 10 10			
9 🗟 📑 💉 🔏	1	Name	Туре	Play Tone						2 2 2 2	н н 1 т
Play Tone		Play Tone 1	Play Tone			1.1				2.2	
Duration (mS)							🥳 F	layF	reque	ncy1💓	
Name Play Tone1							Elem	ents		Out	ا
	-							lay To	one1		
	~										12
						ł	be, Ir				
		2								2012	
						1.1				10 IV	
										10 IV	10 IV
						i - i			10 A.	÷ ÷	
								2 2	10 IV	10 R	2.2
						1 I		2.2		2.12	• •
	8-										• •
	1.1		100			1					
			🗘 🔘 Digi		ut 🔟 🗗 👘 👘						

Preview and Navigation



Above the Component Properties Window, you'll find a preview of the entire project. This preview provides a bird's-eye view of your project's layout. What makes this feature even more useful is that, by moving your mouse over the preview, a green rectangle appears. You can click and drag this rectangle to navigate quickly from one part of the project to another in the main project workspace. This navigation feature streamlines the process of working on different sections of your project, enhancing your overall productivity.

Understanding and utilizing these additional features in the Component Properties Window, as well as the preview with navigation, will help you work efficiently and maintain a well-organized workspace as you design and develop your electronic projects in Visuino.

Arduino Uno R3 🗇 Serial 💎 Serial[0] (9600) ¢∲In 2 Sending 0 Out I2C Channel **F**12C **þ**₿€In Out 1 Request 1 🖁 Digital 558:93 Help 🙀 Libraries С Platforms \sim

Navigating the Visuino Panel

When you first open Visuino, you'll encounter the Visuino interface, which is designed to make your electronic project development as smooth as possible. At the bottom of the interface, you'll find a window with various tabs that serve different purposes. Let's take a closer look at these tabs:

9.1. Bottom Window Tabs

🕜 Help	Region B	uild 👎	🖗 Seria	al 🦥	Errors	🏟 Platforms	🚳 Libraries
e , •		0			8		
					Des	cription	
					Use	his compone	ent to define array of Digital Values.
					Diag	ram:	
						ArduinoDigital	ValueArray
						idex	
					Cat	egories	
					•	Categor	y Data Sources/Digital - Digital Data Source compon-

- **Help:** The Help tab is your source of useful information. It provides you with details about your version, trial version information, or even displays the main wiki page if your computer is connected to the internet. This tab is your gateway to understanding Visuino and its features.
- **Build:** The Build tab is where you'll find functionalities related to compiling/building, and uploading your project. It's an essential part of the development process.
- Serial: The Serial tab allows you to manage the communication between your project and the connected microcontroller. It's where you can view and interact with the serial data sent and received from the microcontroller.
- Errors(Optional): The Errors tab is where any errors or warnings related to your project will be displayed. It helps you identify and resolve issues that may arise during your project

development. This Tab is not shown if there are no errors.

- **Platforms:** The Platforms tab is where you can manage the different microcontroller platforms. Visuino automatically installs the platforms that you need, but you may have to update to a new version from time to time.
- **Libraries:** The Libraries tab allows you to manage the libraries used in your project. You can add, remove, or update libraries to suit your project's needs. Visuino automatically installs the libraries that you need, however you may have to update to a new version from time to time, or add additional libraries that you want to use.
- Button On/Off: This button will show or hide the tabs

9.1.1. Help Built-in Web Browser

The Help built-in web browser, found within the Help tab, is a valuable tool for accessing information related to your Visuino projects.

It provides:

- Important Information: You'll find information about your Visuino version, or trial version details, ensuring you have the necessary information at your fingertips.
- Wiki Page Loading: When working on your projects, this tab also loads a wiki page related to the selected component or microcontroller board. This feature is incredibly useful for understanding the components you're using and making informed decisions during the project development process.

🕜 Help	🆏 Build	💎 Serial	1	Errors 🏾	Platforms	💫 Libraries
e , 🔹	۵ 🖗	💿 🧕	2	9		
				Desci	ription	
				Use this	compone	ent to define array of Digital Values.
				Diagrar	n:	
				TAr	duinoDigital	ValueArray
					k	Out 🔯 🗗
				Categ	gories	
				. 0	Categor	v Data Sources/Digital - Digital Data Source compo

Here are some of its key features:

- **Back and Forward:** The toolbar includes navigation buttons for going back and forward, allowing you to revisit previously viewed pages or navigate through pages you've recently viewed.
- **Reload:** Clicking the reload button refreshes the currently displayed page. This is particularly useful when you want to update the information you're viewing.
- **Stop:** The stop button halts the loading of a page. It's handy when you want to interrupt a page from loading, especially if it's taking too long.
- Live Help (On/Off): The Live Help button is a dynamic feature. When enabled, it loads

information from the Visuino wiki about the selected components and microcontroller boards. This provides real-time assistance as you work on your project.

- **Open in Default Web Browser:** If you prefer to view the loaded wiki page in your default web browser, you can use this button to open it in an external browser.
- **Print to PDF:** To save the loaded wiki page as a PDF document, you can utilize the Print to PDF button. This is helpful when you want to keep offline copies of important information.
- Print: The Print button allows you to send the currently loaded page directly to your printer. It's a quick way to obtain a hard copy of the information you need.

The Help built-in web browser and its toolbar functions are here to make sure you have easy access to useful information, whether it's about components, microcontroller boards, or any other aspect of your Visuino projects. In the following chapters, we'll demonstrate how to use this resource effectively to enhance your project development experience.

9.1.2. Build Tab



The Build tab is your gateway to compiling and uploading your Visuino projects to microcontroller boards. Here's a breakdown of the elements and functions you'll find on this tab:

- Selected Board Box: This box displays the selected microcontroller board for your project. Visuino will automatically select the board based on the hardware you're using. If you prefer to change the board manually, you can do so using the selection box. You also have the option to restore it to the default setting by clicking the "Restore to Default" button next to it.
- Verbose Checkbox: The Verbose checkbox allows you to enable or disable verbose output. When verbose output is enabled, you'll receive more detailed information in the console. This is particularly helpful for troubleshooting errors during the build or upload process.
- **Port Selection Box:** Ensuring that the correct port is selected is essential for successful project uploads. When you connect your microcontroller wit USB to the computer, the list of ports will automatically be updated.
- **Upload Button:** Once you've set your board and port, clicking the "Upload" button initiates the upload process. If the project has not been built, it will be automatically build before uploading. This step takes your project from the visual design stage to the execution stage on the microcontroller board.

The Build tab streamlines the process of preparing and deploying your projects. By understanding how to use these features effectively, you'll be well-equipped to compile, upload, and run your Visuino projects on your chosen microcontroller board.

In the upcoming chapters, we will provide detailed guidance on utilizing the Build tab for a seamless project development experience.

9.1.1.1 Build Tab - Additional Features

- **Refresh Index Button:** On the left side of the Build tab, you'll find the "Refresh Index" button. Clicking this button triggers a refresh of the board indexes from the internet, ensuring you have access to the latest indexes for various microcontroller boards. This step is important for staying up-to-date with the latest board information.
- **Build/Compile Button:** The "Build/Compile" button allows you to compile your project without initiating the upload process. This separation of functions provides flexibility in your project development, enabling you to compile your code independently.
- **Compile/Build and Upload Button:** Clicking the "Compile/Build and Upload" button initiates both the compilation and upload process.
- Autoscroll Checkbox: The "Autoscroll" checkbox in the output window provides control over the scrolling behavior. Enabling this option ensures that the output window automatically scrolls when compiling or uploading, helping you keep track of the process.
- **Timestamp Checkbox:** The "Timestamp" checkbox adds a timestamp to the output window during compiling or uploading. This timestamp serves as a reference point, making it easier to identify when specific actions took place.
- **Open Build Folder Button:** The "Open Build Folder" button opens the folder where your project files are built. This allows you to access the compiled code and related files for further examination or storage.
- Copy Latest Build Hex/Bin File Name Path Button: Clicking this button copies the file name and path of the latest built hex or bin file. This feature simplifies the process of locating and managing the compiled code.

These additional features within the Build tab enhance your control and visibility during project development. Whether you need to refresh board indexes, fine-tune your code, or manage the output, Visuino provides the tools to make your development process more efficient and effective. In the following chapters, we'll dive deeper into using these features to create and deploy your electronic projects.

9.1.3. Serial Tab

The Serial tab in Visuino plays a pivotal role in facilitating communication between your project and the microcontroller board. Here, you'll find an array of settings and options to ensure your serial communication is reliable and tailored to your project's requirements. Let's explore the key features of the Serial tab:

😢 Help 🍇 Build 👻 Serial 👋 Errors 象 Platforms 🖏 Libraries
Port: Upload Port - COM9 🗸 Speed: Auto Config (9600) 🗸 Format: Auto Select (Unformatted 🗸 🖉 Reset 🛛 Log 📄 🥰 Connect 🗆 Connect on Upload
E Terminal Scope
Auto Scroll Hold Word Wrap Special Characters Timestamp Show Output
3 -201
a -918
■ 154 ■ 433
162
-716
50 -683

• **Port Selection Box:** The Port Selection Box provides a means to choose the appropriate communication port for your microcontroller board.

- **Speed Selection Box:** The Speed Selection Box allows you to specify the baud rate for the serial communication. This rate determines the speed at which data is transmitted and received. The speed has to match the speed specified for the serial port in your project. If "Auto Config" is selected, Visuino will automatically set the speed, based on the setting in your project.
- Format Selection Box: The Format Selection Box lets you define the format for serial communication. You can choose the format that aligns with your project's requirements.
- **Reset Checkbox:** Enabling the "Reset" checkbox initiates a reset operation on the microcontroller board, providing a fresh start for your project.(Not all controller types support Reset)
- Log Checkbox: By checking the "Log" checkbox, you can log serial data to file. This feature is useful for recording and reviewing communication data.
- Save Button: The "Save" button allows you to select file to log the serial output.
- **Connect Button:** Clicking the "Connect" button establishes the serial connection between Visuino and the microcontroller board. This is essential for sending and receiving data.
- **Connect on Upload Checkbox:** Enabling the "Connect on Upload" checkbox automates the process of connecting to the serial port after you upload the project to the microcontroller board. This saves you time and ensures that the connection is established promptly.
- **Terminal Tab:** The Terminal Tab provides a dedicated space for interacting with serial data in text form. It allows you to send and receive data and observe the communication in real-time.
- **Scope Tab:** The Scope Tab offers visualization capabilities, allowing you to view the data in a graphical format. This feature is valuable for analyzing data trends and patterns.

The Serial tab empowers you to set up and manage serial communication efficiently, enhancing the functionality and interactivity of your electronic projects. In the following chapters, we'll explore the Serial tab in more detail and demonstrate how to effectively utilize it for your projects.

9.1.3.1. Terminal Tab

The Terminal tab in Visuino is an essential tool for monitoring and interacting with serial data as part of your project development. It provides several features to enhance your experience and control over serial communication. Let's explore the functionalities within the Terminal tab:

Terminal Tab - Features

- Autoscroll Checkbox: The Autoscroll checkbox offers control over the scrolling behavior within the terminal. When enabled, the terminal window automatically scrolls as new data is received, ensuring you can continuously view the latest information.
- Hold Checkbox: The Hold checkbox provides a way to pause the update of the serial terminal window. When enabled, you can temporarily stop the update with new data to examine specific data. This feature is valuable when you want to review or analyze particular information.

- Word Wrap Checkbox: Enabling the Word Wrap checkbox allows text to automatically wrap to the next line within the terminal window. This feature ensures that long lines of text do not extend beyond the visible area, providing a more organized view of your serial data.
- **Special Characters Checkbox:** The Special Characters checkbox allows you to display and interpret special characters within the terminal window. When enabled, control characters and escape sequences are interpreted and presented in a readable format, enhancing your understanding of the data.
- **Timestamp Checkbox:** Enabling the Timestamp checkbox adds timestamps to the data received in the terminal. This is particularly useful for tracking the timing of data transmissions, aiding in analysis and debugging.
- Show Output Checkbox: The Show Output checkbox controls whether the output from the computer to the microcontroller is displayed in the terminal window. When enabled, you can view the project's output in real-time, providing valuable insight into its behavior.

These features within the Terminal tab offer flexibility and control when working with serial data in Visuino. Whether you need to pause, analyze, or format your data, the Terminal tab ensures you have the tools to manage your project effectively. In the following chapters, we'll explore how to utilize these features to streamline your project development.

Scope Tab in the Serial Tab of Visuino

The **Scope Tab** offers powerful visualization capabilities, allowing you to view data in a graphical format. This feature is invaluable for analyzing data trends and patterns, making it easier to understand the behavior of your electronic projects.

Key Features of the Scope Tab:

- **Graphical Data Visualization**: View real-time data in graphs to monitor changes and trends.
- Settings Buttons: Located in the top right corner, these buttons allow you to customize various aspects of the visualization, such as scaling, colors, and display options.
- **Channel Management:** On the right side of the Scope Tab, you can turn on or off channels, which represent different data streams. This flexibility lets you focus on specific data points and reduces clutter by only displaying the information you need.



9.1.4. Errors Tab(Optional)

In addition to designing your electronic projects, Visuino provides various tools and tabs for managing libraries, platforms, and monitoring errors. These functionalities help you ensure that your projects run smoothly and are up-to-date. Let's explore these tabs in detail:

🕜 н	elp 🦓 B	Build errors 👋 Errors	🧒 Plat	atforms 🚯 Libraries
Line 9	Column 6	File C:\Users\B\Documents\Arduin	o¥i	Error Class type "TArduinoTextToJoaat" is registered but has no image!

The Errors Tab is your window to monitoring errors that might occur with the loaded libraries and components. It helps you identify and address issues in your projects during the development process. If there are no errors, this Tab is not visible.

9.1.5. Platforms Tab

Installed	Latest	Name	Package	Maintainer	Boards
1.8.6	1.8.6	Arduino AVR Boards	arduino	Arduino	Arduino Yún, Arc Mega, Arduino Me Arduino Mini, Ardu Arduino ATMegaN Playground, Ardui
	1.8.8	Arduino megaAVR Boards	arduino	Arduino	Arduino UNO WiFi
	1.6.12	Arduino SAM Boards (32-bits ARM Cortex-M3)	arduino	Arduino	Arduino Due
	1.8.13	Arduino SAMD Boards (32-bits ARM Cortex-M0+)	arduino	Arduino	Arduino MKR WiFi Arduino MKR WAN
a *					

The Platforms Tab offers insight into the installed and supported microcontroller boards and platforms. Here's a detailed breakdown of the features within the Platforms Tab:

- **Installed Version:** This displays the currently installed version of each platform or microcontroller board.
- Available Latest Version: Visuino keeps you informed about the latest available versions of platforms and boards. To access the latest versions, you can click on the "Refresh Index" button. This action refreshes the platforms' indexes from the internet and downloads the latest data.
- Install/Update and Uninstall: To install or update a platform or board, simply right-click

with your mouse and select "Install/Update." Conversely, to uninstall a platform or board, right-click and choose "Uninstall."

Filter: All		~
Installed	Latest	Name
1.8.6	1.8.6	Arduino AVR Boards
		😝 Install/Update
	1.8.8	Arduino megaAVR Boards
	1.6.12	Arduino SAM Boards (32-bits ARM Cortex-M3)
	1.8.13	Arduino SAMD Boards (32-bits ARM Cortex-M0+)

• **Changing Versions:** If you wish to switch to another version of a platform or board, click with your mouse on the version number. A window will open, allowing you to select the desired version for installation.



- **Tip:** When encountering errors during project compilation or upload, it's advisable to first check for a newer version. Outdated platforms or boards can often be the root cause of these errors.
- Filter Box: You can utilize the Filter box to sort and display platforms based on whether they are updatable, installed, not installed, or by using specific keywords for filtering.

Help	©⊜ B	uild ອ Serial 👋 Errors	🏟 Platforms	🚯 Libraries	
Filter: All		V ESP32			
Installed	Latest	Name	Package	Maintainer	Boards
	2.0.10	esp32	arduino	Arduino	Arduino Nano ESP32
	2.0.13	Arduino ESP32 Boards	arduino	Arduino	Arduino Nano ESP32
2.0.10	2.0.11	esp32	esp32	Espressif Systems	ESP32 Dev Board, ESP32-S2 Dev Board
	0.0.6	FireBeetle-ESP32 Mainboard	esp32	DFRobot DFRDuino	FireBettle-ESP32
	2.1.4	industrialshields-esp32	industrialshields		WiFi/BT module, 10 IOs PLC family, ES

9.1.6. Libraries Tab

Installed	Latest	Name	Architectures
	1.7.0	Bridge	
	2.5.9	Firmata	
2.0.2	2.0.2	Ethernet	
	1.0.4	Robot Control	avr
	2.0.0	Robot IR Remote	avr
	1.0.3	Robot Motor	avr

The Libraries Tab serves a similar purpose, providing information about installed libraries. Here's a breakdown of its features:

- Installed Version: The installed version of each library is displayed for your reference.
- Available Latest Version: Visuino keeps you informed about the latest versions of libraries. You can access this information by clicking the "Refresh Index" button, which updates the libraries' indexes from the internet and downloads the latest data.
- Install/Update and Uninstall: To install or update a library, right-click with your mouse and select "Install/Update." To uninstall a library, right-click and choose "Uninstall."

Filter: All	~			
Installed	Latest	Name		
	1.7.0	Bridge		
	2.5.9	Firmata		
2.0.2	2.0.2	Ethernet		
	1.0.4	Robot Control		
	2.0.0	Robot IR Remo		
	1.0.3	Robot Motor 🙀 Uninstall		
<				

• **Changing Versions:** If you want to change the version of a library, click on the version number. A window will open, enabling you to select the desired version for installation.

ilter: All	~		(Uninstall) 2.0.2	^
Installed	Latest	Name	2.0.1	
	1.7.0	Bridge	2.0.0	
	2.5.9	Firmata	1.1.1	
2.0.2	2.0.2	Ethernet	1.1.0	
	1.0.4	Robot Control	1.0.5	
	2.0.0	Robot IR Remot	e 1.0.3	
	1.0.3	Robot Motor	107	~

- **Tip:** When you encounter errors during project compilation or upload, consider checking for a newer library version. This simple step can often resolve the issues you face.
- Filter Box: Similar to the Platforms Tab, the Filter box in the Libraries Tab allows you to sort and display libraries based on their update status, installation status, or by using specific keywords for filtering.

By effectively using the Errors Tab, Platforms Tab, and Libraries Tab, you can ensure that your projects are free from errors and are always running with the latest libraries and platform versions. These features are indispensable for maintaining your project's integrity and performance. In the following chapters, we'll delve deeper into using these tabs and options to enhance your project development experience.

Advanced Features in Visuino

10.1. Instruments Tab

Visuino instruments, are virtual tools within the Visuino software environment that allow users to visualize and interact with data from sensors and other components connected to an Arduino board. These instruments are particularly useful for scenarios where an external display is unavailable or when performing calculations and experiments that require real-time data monitoring.



To use <u>Instruments</u> in Visuino first Add "**Packet**" component from the component list and connect "**Packet**" component pin [Out] to the Arduino board Serial Pin. Double click on the "**Packet**" component to Add & Set the Input Elements. For Each element you can set different type of the instrument in the properties window.





10.2. Advanced Features in the Professional Version

In the Professional version of Visuino, you gain access to a host of advanced features that take your project development to the next level. One such feature is the extra tab within the main window, which allows you to view the live code generated by your project. Let's explore this feature and the additional options available through the live code tab:

10.2.1. Live Code Tab

In the Pro version, the main window includes an extra tab dedicated to viewing the live code generated by your project. This tab offers a real-time glimpse into the code that your project is producing as you design it. It's a powerful feature for those who want to dive into the details of their project's code and fine-tune it.

```
Main 📑 Code
 1 //-
 2 //
 3 //
            Sketch Generated by Visuino
 4 //
            www.visuino.com
 5 //
              Version 8.0.0.94
 6 //
 7 //-
 8
 9 #define VISUINO ARDUINO UNO
10
11 #include <OpenWire.h>
12 #include <Mitov.h>
13 #include <Mitov StandardSerial.h>
14 #include <Mitov ValueArray.h>
15 #include <Mitov RandomGenerator.h>
16 #include <Mitov Timing.h>
17
18
19 // Arduino Constant Declarations
20
21 namespace VisuinoConstants
22 {
23 class FloatValue0
24 {
25 public:
     inline static constexpr float GetValue() { return 1; }
26
```

10.2.2. Contextual Menu for Live Code

Right-clicking within the live code tab opens a contextual menu that provides access to a range of code-related functions:

- Find: Open a search window at the bottom with options to search through the code, including checkboxes to select "Ignore case," "Whole words only," and "Use regular expression."
- Find Next: Move to the next search result in the code.
- Find Previous: Move to the previous search result.
- Go to Line: Open a window where you can enter a line number to jump to a specific part of the code.
- Select All: Highlight and select the entire code for easy copying.
- Copy: Copy the selected code to your clipboard for external use.
- **Refresh Code:** Visuino automatically refreshes the code. This option is rarely needed, and you can ignore it. It is used by the developers of Visuino during our testing.

These features within the live code tab and contextual menu in the Professional version give you an unprecedented level of control and insight into your project's code, enabling you to fine-tune your project with precision.

In the following sections, we will delve deeper into these advanced features and demonstrate how to harness their capabilities to create and manage your electronic projects effectively.
Understanding Analog Values in Visuino

Analog values in Visuino, like in many other digital and microcontroller systems, represent continuous data. Unlike digital values which are either on (1) or off (0), analog values can vary smoothly over a range. In Visuino, these values are normalized to fall within a specific range for simplicity and consistency.

The Range of Analog Values

In Visuino, analog values are scaled to fall within the range of 0 to 1. This means that:

0 represents the minimum possible value of the analog signal.

1 represents the maximum possible value of the analog signal.

Values such as 0.1, 0.25, 0.5, etc., represent intermediate levels between the minimum and maximum.

By normalizing the values to a range of 0 to 1, Visuino simplifies the programming and processing of these values. This allows developers to work with a consistent and predictable range regardless of the actual hardware specifications.

Compatibility: Normalizing to a 0-1 range ensures that the analog values are compatible with a variety of components and modules within the Visuino environment, which may have different native ranges. The main reason for normalizing is to make sensors and actuators compatible - using the same range, as example Potentiometer and Servo.

Ease of Scaling: It is easier to scale these values up or down as needed for specific applications or when interfacing with different sensors and actuators.

Using If/Else in Visuino with Compare components

Visuino enables conditional logic using Compare components, similar to If/Else statements in traditional programming. These components evaluate conditions and output True or False based on the result. Common Compare components include: Compare Analog Value, Compare Integer Value, Compare Text Value, and so on.

These are just a few examples. Each Compare component allows you to implement precise conditional logic tailored to your project's needs. Using these components, you can create complex behaviors and interactions in your Visuino projects.

Implementing While with Sequence component

In Visuino, the "Sequence" component can be used to create loops similar to the "while" loop in traditional programming. The Sequence component allows you to execute a set of actions repeatedly based on specific conditions.

Implementing For Loops with Repeat components

For loop-like behavior can be achieved using Repeat components. These components allow you to repeat a set of actions a specified number of times or until a condition is met.

Implementing Switch Logic

There are many components that do Switch work, they tend to be Switch, but also Binary Decoder, Arrays and many others.

Poject Examples

11.1. LED Blink



In this Tutorial we will learn how to make a Simple LED Blink project using Arduino board.

Step 1: What You Will Need

- Arduino UNO, Arduino Mega, or any other board supported by Visuino.
- Visuino program





Step 2: Start Visuino, and Select the Arduino Board Type

Start Visuino as shown on the picture. By default Visuino will have Arduino UNO R3 selected as board. If you want to use different board, click on the "Tools" button on the Arduino component in Visuino When the dialog appears, select your board as shown on the picture.





Step 5: In Visuino Add, Set & Connect Components



On the right side of Visuino type "**Pulse Generator**" in the search box and drag the component to the main Area

• Connect "**PulseGenerator1**" component Pin [Out] to "Arduino" Digital Pin [13] (You may need to connect to a different pin depending on what pin your board has LED)



Step 6: Connect your Arduino board to the computer

Connect your Arduino board to the computer with USB cable.

Step 7: Generate, Compile, and Upload the Arduino Code

In Visuino, at the bottom click on the "Build" Tab, make sure the correct port is selected, then click on the "Upload" button.

1216	373
0	Help 🎭 Build 🤝 Serial 😻 Errors 🐠 Platforms 職 Libraries
*	Board: Arduino Uno 🔗 🗋 Verbose 🗞 Port: COM10 🧹 🗍 Save 🗋 Verbose 🐗 🐗 🎢 🖓 Auto Scroll 🗋 Timestamp 🞼 🎼
	Building compile -b arduino:avr:unoexport-binariesno-color "C:\Users\B\AppData\Local\Temp\VisuinoBuild47052\Generated\ Sketch uses 444 bytes (1%) of program storage space. Maximum is 32256 bytes. Global variables use 9 bytes (0%) of dynamic memory, leaving 2039 bytes for local variables. Maximum is 2048 bytes.
80	Used library Version Path
*	Visuino 8.0.0-120 C:\Users\B\Documents\Arduino\libraries\Mitov
86	Used platform Version Path
	arduino:avr 1.8.6 C:\Users\B\AppData\Local\Arduino15\packages\arduino\hardware\avr\1.8.6 Buind completed - Time: 00:00:02 Uploading

Play

If you power the Arduino board, the LED will start to blink. Congratulations! You have completed your project with Visuino

Weather Station Using Arduino UNO & Visuino

In this tutorial we will learn how to make a simple weather station using the Arduino UNO board & Visuino to display the Temperature & Humidity values on the Visuino Instruments.



Step 1: What You Will Need

- Arduino UNO
- DHT11 sensor
- Jumper wires
- Visuino program: Download Visuino









Step 2: The Circuit

- Connect DHT11 Sensor pin (VCC +) to Arduino pin [+5V]
- Connect DHT11 Sensor pin (GND -) to Arduino pin [GND]
- Connect DHT11 Sensor pin (S) to Arduino digital pin (2)



Step 3: Start Visuino, and Select the Arduino UNO Board Type

Start Visuino as shown on the picture. By default Visuino will have Arduino UNO R3 selected as board. If you want to use different board, click on the "Tools" button on the Arduino component in Visuino When the dialog appears, select your board as shown on the picture





Step 4: In Visuino Add Components

• Add "DHT11" component



• Add "Packet" component

Step 5: In Visuino Set Components

Select the "Packet1" component In the Object Inspector expand the "Head Marker"property In the Object Inspector click on the "..." button In the Bytes editor type some numbers, as example 55 55





Click on the OK button to confirm and close the editor

Double click on the "Packet1" component and in the "Elements" window drag two times "Analog" element to the left side

Optionally you can select the type of the Instrument for each element in the properties window, where you can also set the Scale.





Step 6: In Visuino Connect Components

- Connect "HumidityThermometer1" pin [Temperature] to "Packet1">"Analog1" pin [In]
- Connect "HumidityThermometer1" pin [Humidity] to ""Packet1">"Analog2" pin [In]
- Connect "HumidityThermometer1" pin [Sensor] to Arduino digital pin [2]
- Connect "Packet1" pin [Out] to Arduino board Serial pin [In]



Step 7: Generate, Compile, and Upload the Arduino Code

In Visuino, at the bottom click on the "Build" Tab, make sure the correct port is selected, then click on the "Upload" button.

1216	373
0	Help 🎭 Build 💗 Serial 😻 Errors 🐠 Platforms 🖏 Libraries
*	Board: Arduino Uno 🕢 🗌 Verbose 🐁 Port: COM10 🧹 🗋 Save 🗋 Verbose 🐗 🐗 🏹 🗹 Auto Scroll 🗌 Timestamp 🎼 🔞
	Building compile -b arduino:avr:unoexport-binariesno-color "C:\Users\B\AppData\Local\Temp\VisuinoBuild47052\Generated\ Sketch uses 444 bytes (1%) of program storage space. Maximum is 32256 bytes. Global variables use 9 bytes (0%) of dynamic memory, leaving 2039 bytes for local variables. Maximum is 2048 bytes.
86	Used library Version Path
*	Visuino 8.0.0-120 C:\Users\B\Documents\Arduino\libraries\Mitov
84	Used platform Version Path
	arduino:avr 1.8.6 C:\Users\B\AppData\Local\Arduino15\packages\arduino\hardware\avr\1.8.6 Buind completed - Time: 00:00:02 Uploading

Step 8: Play

If you power the Arduino module, select Serial tab> Instruments, Select the correct Port and click connect button & Arduino Display will start to display the Temperature and Humidity values in the Visuino Instruments.

Congratulations! You have completed your project with Visuino.



11.3 How to Use I2C OLED Display



In this tutorial we are going to Print a text on the OLED Display using Arduino.

Step 1: What You Will Need

• Arduino UNO (or any other board)





- Jumper wires
- Breadboard
- Visuino program: <u>Download Visuino</u>







Step 2: The Circuit

- Connect OLED Display pin [SCL] to Arduino pin [SCL]
- Connect OLED Display pin [SDA] to Arduino pin [SDA]
- Connect OLED Display pin [VCC] to Arduino pin [5v]
- Connect OLED Display pin [GND] to Arduino pin [GND]



Step 3: Start Visuino, and Select the Arduino Board Type

Start Visuino as shown on the picture. By default Visuino will have Arduino UNO R3 selected as board. If you want to use different board, click on the "Tools" button on the Arduino component in Visuino When the dialog appears, select your board as shown on the picture





Step 4: In Visuino Add & Set Components



- Add "OLED I2C" component
- Double click on the "**DisplayOLED1**" and in the "**Elements**" window drag "**Draw Text**" to the left side and set in the properties window set "**Size**" to 2, "**Text**" to HELLO (Or any other text)





- **Optionally** you can also set X and Y
- Close the "Elements" window
- Connect "DisplayOLED1" I2C pin [Out] to Arduino board I2C pin [In]



Step 5: Generate, Compile, and Upload the Arduino Code

In Visuino, at the bottom click on the "**Build**" Tab, make sure the correct port is selected, then click on the "Upload" button.

1216	373
0	Help 🎭 Build 💗 Serial 😻 Errors 🐠 Platforms 🖏 Libraries
*	Board: Arduino Uno 🔗 🛛 Verbose 🐁 Port: COM10 🧹 🗋 Save 🗋 Verbose 🐗 🐗 🌠 🗹 Auto Scroll 🗋 Timestamp 🎼 🔞
*	Building compile -b arduino:avr:unoexport-binariesno-color "C:\Users\B\AppData\Local\Temp\VisuinoBuild47052\Generated\ Sketch uses 444 bytes (1%) of program storage space. Maximum is 32256 bytes. Global variables use 9 bytes (0%) of dynamic memory, leaving 2039 bytes for local variables. Maximum is 2048 bytes. Used library Version Path
80	Visuino 8.0.0-120 C:\Users\B\Documents\Arduino\libraries\Mitov
2	Used platform Version Path
*	Buind completed - Time: 00:00:02 Uploading

Step 6: Play

If you power the Arduino module, The OLED Display will show the Text. Congratulations! You have completed your project with Visuino.

Ultrasonic Range Finder



In this tutorial we will learn how to use Ultrasonic Sensor HC-SR04 & Arduino to display the obstacle distance in the Visuino serial window.

Step 1: What You Will Need

- Ultrasonic Sensor HC-SR04
- Arduino UNO (or any other board)
- Jumper wires
- Visuino program: Download Visuino









Step 2: The Circuit

- Connect Ultrasonic module pin (VCC) to Arduino pin [+5V]
- Connect Ultrasonic module pin (GND) to Arduino pin [GND]
- Connect Ultrasonic module pin (ECHO) to Arduino pin digital (2)
- Connect Ultrasonic module pin (TRIG) to Arduino pin digital (3)



Step 3: Start Visuino, and Select the Arduino Board Type

Start Visuino as shown on the picture. By default Visuino will have Arduino UNO R3 selected as board. If you want to use different board, click on the "Tools" button on the Arduino component in Visuino When the dialog appears, select your board as shown on the picture





Step 4: In Visuino Add & Connect Components

• Add "Ultrasonic Ranger(Ping)" component



- Connect "UltrasonicRanger1" pin [Echo] to Arduino digital pin [2]
- Connect "UltrasonicRanger1" pin [Ping-Trigger] to Arduino digital pin [3]



• Connect "UltrasonicRanger1" pin [Out] Arduino Serial pin [In]

Step 7: Generate, Compile, and Upload the Arduino Code

In Visuino, at the bottom click on the "Build" Tab, make sure the correct port is selected, then click on the "Upload" button.

1216:37	73						
🕜 He	lp 🎭 Build 💗 Serial 😻 Errors 🐠 Platforms 🙀 Libraries						
🤣 в	oard: Arduino Uno 🦳 🕢 🖓 🗌 Verbose 🐁 Port: COM10 🧹 🗋 Save 🗌 Verbose 🐗 🐗 🎢 🖓 Auto Scroll 🗌 Timestamp 🔯 👔						
	Building compile -b arduino:avr:unoexport-binariesno-color "C:\Users\B\AppData\Local\Temp\VisuinoBuild47052\Generated\ Sketch uses 444 bytes (1%) of program storage space. Maximum is 32256 bytes. Global variables use 9 bytes (0%) of dynamic memory, leaving 2039 bytes for local variables. Maximum is 2048 bytes.						
8.	Used library Version Path						
%	Visuino 8.0.0-120 C:\Users\B\Documents\Arduino\libraries\Mitov						
80	🐁 Used platform Version Path						
80	arduino:avr 1.8.6 C:\Users\B\AppData\Local\Arduinol5\packages\arduino\hardware\avr\1.8.6						
84	Buind completed - Time: 00:00:02						
۲	Uploading						

Step 8: Play

If you power the Arduino module and click connect in the "Serial" tab, you will be able to see the Obstacle distance measured by ultrasonic sensor.

Congratulations! You have completed your project with Visuino.

🕜 He	elp 🦏 Build	🛡 Sei	rial 👋	Errors	😸 Platform	is 🙀 Libi	raries				
Port:	COM8 🗸 S	peed: Au	ito Config	(9600)	Eormat:	Auto Sele	ct (Unformatted	Reset (D	rR) Log	📄 🞽 Disconr	nect Conn
Т	erminal 🔛	Scope									
🗹 A.	uto Scroll	Hold	Word	Wrap	Special C	haracters	Timestamp	Show Outpu	t		
🦈 1	4.40										
🤹 1	4.47										
🦈 1	4.82										
🤹 1	4.66										
🤹 1	4.70										
- 1	4.52										
- 1	4.43										
a 1	4.78										

How to Save Sensor Data Temp & Time to SD Card



In this tutorial we are going to save Temperature and Humidity data from the Dht11 sensor to the SD card and we will also add a time and a date that we will get from the RTC module ds1307.

Step 1: What You Will Need

- Arduino UNO (or any other Arduino or ESP) •SD Card module + SD card to store data •DHT11 Temperature & Humidity sensor •RTC DS1307 module
- •Jumper wires
- •Breadboard
- •Visuino program













Step 2: The Circuit



- Connect SD card Module pin GND to Arduino pin GND
- Connect SD card Module pin VCC to Arduino pin 5V
- Connect SD card Module pin SCK to Arduino digital pin 13
- Connect SD card Module pin MISO to Arduino digital pin 12
- Connect SD card Module pin MOSI to Arduino digital pin 11
- Connect SD card Module pin CS to Arduino digital pin 10
- Connect RTC DS1307 module pin[VCC] to Arduino pin[5V]
- Connect RTC DS1307 module pin[GND] to Arduino pin[GND]
- Connect RTC DS1307 module pin[SDA] to Arduino pin[SDA]
- Connect RTC DS1307 module pin[SCL] to Arduino pin[SCL]
- Connect DHT11 sensor pin [VCC] to Arduino pin [5V]
- Connect DHT11 sensor pin [GND] to Arduino pin [GND]
- Connect DHT11 sensor pin [OUT] to Arduino Digital pin [2]

Step 3: Start Visuino, and Select the Arduino Board Type

Start Visuino as shown on the picture. By default Visuino will have Arduino UNO R3 selected as board. If you want to use different board, click on the "Tools" button on the Arduino component in Visuino When the dialog appears, select your board as shown on the picture



6	BoardType	- 0	×
Eilter:			9
Arduino 101 Arduino Due Arduino Esplora Arduino Esplora Arduino Esplora Arduino Elioperato Arduino Liopado Arduino LilyPad Arduino LilyPad Simple Arduino Miga 2560 Arduino Micro Arduino Micro Arduino Micro Arduino Nano Arduino Pro 168 Arduino Pro 328 Arduino Pro Micro			^
Arduino Pro Vini Arduino Yun Bluno Beetle Controllino MAXI Controllino MAXI Automatio Controllino MEGA Controllino MINI Digispark Digispark Pro Digital Loggers PLDuino ESP32 Development Board	n		~
	✓ 0	K 🗙 Ca	ancel

Step 4: In Visuino Add Components



- •Add "Clock Generator" component
- •Add "Date/Time Value" component
- •Add "Real Time Clock DS1307" component
- •Add "DHT11" component
- •Add "Formatted Text" component
- •Add "Micro SD Card Module" component



Step 5: In Visuino Set Components

			Ele	ments -	. 🗆 🗙
Properties		19 · 19 · 10	🙀 le 🗣 🚄		9
Date/Time Value	P P	Name ABC TextElem AnalogEle AnalogEle	Type TArduinoFormattedT TArduinoFormattedT TArduinoFormattedT	- ABC Text Element 	
Properties					
	9 🐚 📴 🏓 🔏	J		1	
Formatted Text					
 Miscellane Elements Name Name Market 	3 Items FormattedText1 DateTime:%0 TEMP:%1 HUM:%2				

- Select "DateTimeValue1" and in the properties window select "Value" and set your current Date & Time
- Select "ClockGenerator1" and in the properties window set "Frequency" to 0.1 this means





that the Data will be stored every 10 s

- Double click on the "FormattedText1" and in the "Elements" window drag "Text Element" & 2X "Analog Element" to the left side.
- Close the "Elements Window"
- Select "FormattedText1" and in the "properties window select "Text" and type DateTime:

%0 TEMP:%1 HUM:%2

- Double click on the "SDCard1" and in the "Elements" window drag "File" to the left side.and in the properties window select "Path Name" and type SENSDATA.txt
- Close the "Elements Window"

Step 6: In Visuino Connect Components

- Connect "DateTimeValue1" pin [Out] to "RealTimeClock1" pin [Set]
- Connect "HumidityThermometer1" pin [Temperature] to "FormattedText1 > AnalogElement1" pin [In]
- Connect "HumidityThermometer1" pin [Humidity] to "FormattedText1 >AnalogElement2" pin [In]
- Connect "RealTimeClock1" pin [Our] to "FormattedText1 > Text Element1" pin [In]
- Connect "ClockGenerator1" pin [Our] to "FormattedText1" pin [Clock]
- Connect "RealTimeClock1" pin [I2C] to "Arduino" pin [I2C]
- Connect "HumidityThermometer1" pin [Sensor] to "Arduino" Digital pin [2]
- Connect "FormattedText1" pin [Out] to "SDCard1" pin [In]
- Connect "SDCard1" pin [SPI] to "Arduino" pin [SPI]
- Connect "SDCard1" pin [Chip Select] to "Arduino" Digital pin [10]



Step 7: Generate, Compile, and Upload the Arduino Code

In Visuino, at the bottom click on the "Build" Tab, make sure the correct port is selected, then click on the "Upload" button.



Step 9: Play

If you power the Arduino module, The Data from the sensor DHT11 and time from the RTC module DS1307 will be saved every 10s to the SD card.

Congratulations! You have completed your project with Visuino.

Arduino Ethernet Shield & MAX7219 8-digit LED Display - Get Time and Date From the Internet



In this tutorial we will learn how to get the date and time from NIST TIME server using Arduino Ethernet Shield and Display it on the MAX7219 8-digit LED Display.

Step 1: What You Will Need



- Arduino UNO (or any other Arduino)
- Jumper wires
- Ethernet Shield for Arduino
- Max7219 Led Dot Matrix 8-digit Digital Display Control Module
- LAN Internet connection
- Visuino program

Step 2: The Circuit



- Connect Internet LAN cable to the Arduino Ethernet Shield
- Connect **LED module** pin[VCC] to **Arduino Ethernet** Shield pin[5V]
- Connect LED module pin[GND] to Arduino Ethernet Shield pin[GND]
- Connect LED module pin[DIN] to Arduino Ethernet Shield digital pin[11]
- Connect LED module pin[CS] to Arduino Ethernet Shield digital pin[2]
- Connect LED module pin[CLK] to Arduino Ethernet Shield digital pin[13]

Step 3: Start Visuino, and Select the Arduino Board Type

Start Visuino as shown on the picture. By default Visuino will have Arduino UNO R3 selected as board. If you want to use different board, click on the "Tools" button on the Arduino component in Visuino When the dialog appears, select your board as shown on the picture



Step 3: In Visuino Add Ethernet Shield

🍟 Arduino Uno	🤾 🗢 🔍	
it Serial		· · · · · · · · ·
I2C Channels		
😵 In Requ	Select Board	
ुर्दू Digital २२ Digital(RX)[0]	Add Multiple Co Replace Selecte	omponents
Digital(TX)[1]	Grid	Ctrl+G
ि∰ Digital[2] ∭Digital	Arrange Filters	Ctn+A
Digital [3]	Eind	Ctrl+F
Digital [4] (Micro SD (CS))	 Find Next Find Previous 	F3 Shift+F3
Digital [5]	Cut	Ctrl+X Ctrl+C
Analog (PWM)	Paste	Ctrl+V
Analog (PWM)	X Delete Selected	
Digital () Digital () Digital ()		· · · · · · · · ·
Digital[9]		

- Right mouse click on the Arduino board and select "Add Shields"
- In the "Shields" window drag "Ethernet Shield" to the left side and In the properties window expand the "Modules" property, then its "Ethernet" sub-property, and set the value of the "Mac Address" sub-property to some MAC address. In our example "DE-AD-BE-EF-FE-ED", but you should probably use a MAC generator.

🍃 Properties					
	9 🕲 🐮 📑 💉 🔏				
Ethernet Shield					
😑 🃢 Miscellaneous					
📌 Enabled	✓ True				
🖽 🧑 IPAddress	(DNS=(Enabled=False,Gate				
🌛 🖽 🚮 🔨 🧱 Mac Address	DE-AD-BE-EF-FE-ED				
🖽 🔛 Micro SD	(Elements=0 Items,Enabled=				
Operations	0 Items				
Sockets	0 Items				
😹 Use DHCP	True				

• In the properties window select "**Sockets**" and click on the 3 dots button.

• In the "Sockets" window drag "TCP/IP Client" to the left side and in the properties window set "Port" to 37 and "Host" to: time-c-g.nist.gov



- Close the "Sockets" and "Shields" window
- Note: You can find more Time servers here: <u>https://tf.nist.gov/tf-cgi/servers.cgi</u>

ockets	– 🗆 ×
G • G • 🕲 🙀 🗣 🔶	9 😵 123
Name Type	TCP/IP Server



Step 4: In Visuino Add Components

- Pulse Generator
- Internet Time Protocol
- Decode(Split) Date/Time
- 2X "Split Integer Digits
- Maxim LED Display Controller SPI MAX7219/MAX7221











Step 5: In Visuino Set Components





Double click on the "LedController1" and in the "PixelGroups window" drag "Integer Display 7 Segments" to the left side and in the properties window set "Count Digits" to 2, "Leading Zeroes" to True, "Reversed Order" to True

- In the "**PixelGroups window**" drag "**Value Section 7 Segments**" to the left side and in the properties window set "**Initial Decmal Point Value**" to True
- In the "PixelGroups window" drag another "Value Section 7 Segments" to the left side
- In the "**PixelGroups window**" drag "**Integer Display 7 Segments**" to the left side and in the properties window set "**Count Digits**" to 2, "**Leading Zeroes**" to True, "**Reversed Order**" to True
- In the "**PixelGroups window**" drag "**Value Section 7 Segments**" to the left side and in the properties window set "**Initial Decmal Point Value**" to True
- In the "PixelGroups window" drag another "Value Section 7 Segments" to the left side
- Close the "PixelGroups window"

Step 6: In Visuino Connect Components



- Connect "PulseGenerator1" pin [Out] to "InternetTime1" pin [In]
- Connect "InternetTime1" pin [Socket] to "Arduino Uno" >TCP Client1 pin [In]
- Connect "InternetTime1" pin [Out] to "DecodeDateTime1" pin [In]
- Connect "DecodeDateTime1" pin [Month] to "LedController1" > "Integer Display 7 Segments1"pin [In]
- Connect "DecodeDateTime1" pin [Day] to "SplitIntegerDigits1" pin [In]
- Connect "SplitIntegerDigits1" pin[0] to "LedController1" > "Value Section 7 Segments1"pin [In]
- •
- Connect "SplitIntegerDigits1" pin[1] to "LedController1" > "Value Section 7 Segments2"pin [In]
- Connect "DecodeDateTime1" pin [Hour] to "SplitIntegerDigits2" pin [In]
- Connect "SplitIntegerDigits2" pin[0] to "LedController1" > "Value Section 7 Segments3"pin [In]

- Connect "SplitIntegerDigits2" pin[1] to "LedController1" > "Value Section 7 Segments4"pin [In]
- Connect "DecodeDateTime1" pin [Minute] to "LedController1" > "Integer Display 7 Segments2"pin [In]
- Connect "LedController1" pin Out SPI to Arduino Pin SPI [In]
- Connect "LedController1" pin Chip Select to Arduino Digital Pin [2]

Step 7: Generate, Compile, and Upload the Arduino Code

In Visuino, at the bottom click on the "Build" Tab, make sure the correct port is selected, then click on the "Upload" button.



Step 8: Play

If you power the **Arduino**, it will connect to the internet and the display should start showing the date and time from the NIST server.
Visuino Resources

Official Visuino Website

https://www.visuino.com

Official Visuino European Website https://www.visuino.eu

Visuino Wiki (Comprehensive Information) https://visuino.com/wiki

Visuino Examples and Sample Projects https://www.visuino.com/visuino-projects

<u>Visuino Download Page</u>

https://www.visuino.com/downloads

Visuino Third Party Components

https://www.visuino.com/third-party-visuino-components/

Visuino Contact Page https://www.visuino.com/contact

<u>Visuino Facebook Group</u>

https://www.facebook.com/groups/861801593868581/

Visuino MeWe Group

https://mewe.com/join/visuino-visualarduinodevelopers

Visuino on X (Twitter) https://twitter.com/visuino

Visuino on Instructables https://www.instructables.com/search/?q=visuino&projects=all

<u>Visuino on Hackster</u> https://www.hackster.io/search?i=projects&g=visuino

Visuino on electro{maker}

https://www.electromaker.io/search?query=visuino