

Meet Visuino

The image displays the Visuino software interface, titled "Visuino - Visual Arduino Programming (Test55)". The main workspace shows a visual programming diagram for an Arduino Uno. The diagram includes several components: SineGenerator1, SineGenerator2, SineGenerator3, SineGenerator4, PulseGenerator1, and PulseGenerator2. These are connected to a central "Packet1" component, which is then connected to the "Arduino Uno" component. The Arduino Uno component is shown with its digital and analog pins connected to the Packet1 component. The interface also features a component palette on the left, a properties panel, and a bottom panel with gauges and a terminal.

The bottom panel displays several gauges and a terminal window. The gauges are labeled as follows:

- BinaryAnalog1
- BinaryAnalog2
- Digital(Binary)1
- Analog(Binary)1
- Analog(Binary)2
- Digital(Binary)2
- Digital(Binary)3
- Digital(Binary)4

The terminal window shows the following text:

```
648:105 Arduino:Arduino 592:80 Size 144:800  
Port: COM3 Speed: 9600 Format: Packet1
```



Why Arduino for Education?

- Created and widely used by the educational community
- Low cost Open Source Platform
- Resilient difficult to damage hardware
- Huge number of clones and boards new coming every day
- Simple to use for hardware designs
- Very good for real time tasks

And more!

- Huge community
- Very good for artistic projects
- Adopted by developers, students, artists, hobbyists and more, well beyond the traditional HW market
- Huge number of ready to use peripherals

What are the Arduino issues?

- Very primitive and cumbersome development tools, difficult to learn and use
- Requires relatively low level programming
- Most people have no problem connecting hardware to it, but they get lost in the programming side
- Lack of easily available debugging tools
- When used for collecting data, there is no easy way to visualize it

Visuino comes to play

The image displays the Visuino - Visual Arduino Programming interface. The main workspace shows a block-based circuit diagram connected to an Arduino Uno. The circuit includes the following components:

- Arduino Ultrasonic Ranger 1**: Connected to Digital pins 12 and 13. Its Echo pin is connected to Digital pin 12, and its PingOutputPin is connected to Digital pin 13.
- ArduinoDivideByValue 1**: Takes an input from the Ultrasonic Ranger's Echo pin and outputs to the Servo's In pin.
- ArduinoMultiplyByValue 1**: Takes an input from the Ultrasonic Ranger's Echo pin and outputs to the Servo's In pin.
- ArduinoServo 1**: Receives input from the division and multiplication blocks and is connected to Digital pin 9.
- ArduinoStepperMotor4Wire 1**: Connected to Digital pins 2, 3, 4, and 5.

The interface also features a Properties panel on the left, a Terminal and Scope window at the bottom, and a component palette on the right. The Scope window shows a square wave signal on Channel 0, with the Y-axis ranging from 0 to 3500 and the X-axis (Samples) ranging from 3850 to 4850.

What is Visuino

- Graphical development environment for Arduino
- Automatically generates Arduino code, and programs the boards
- Built in data visualization
- Direct mapping of software and hardware components

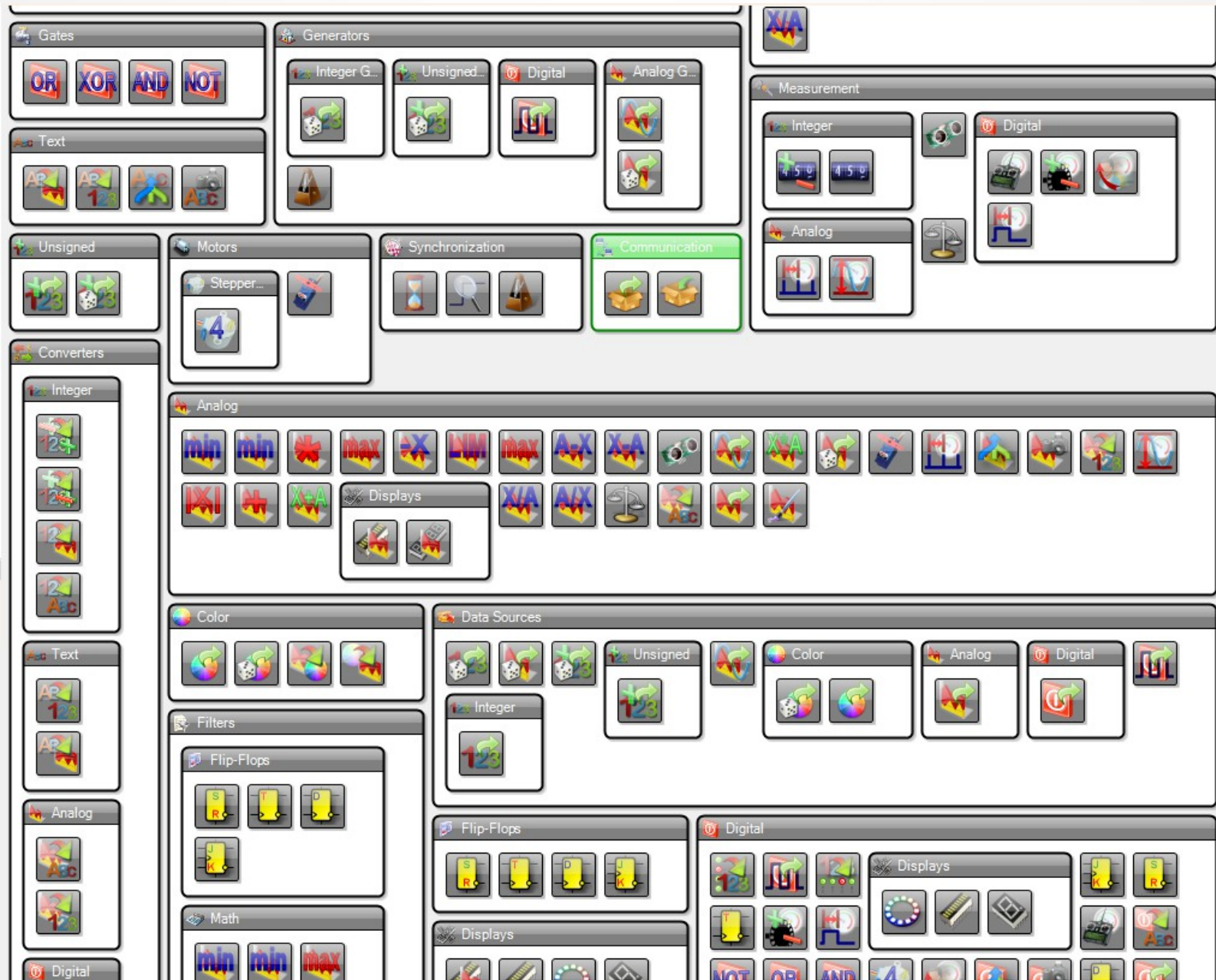
Why Visuino?

- Does not require any programming knowledge
- Simple and intuitive to use even for kids
- Removes the dangers of coding bugs
- Eliminates the need of code debugging
- Well suited for people with hardware knowledge
- Very complex designs can be created and deployed in minutes
- Automatic hardware configuration
- Integrates terminal, scope and visual instrumentation

And more!

- Rich, flexible and expandable visual component framework with open API
- Generates small and highly efficient code
- Direct representation of hardware components in the software side
- Fast growing user community of over 3000 members

Rich component palette

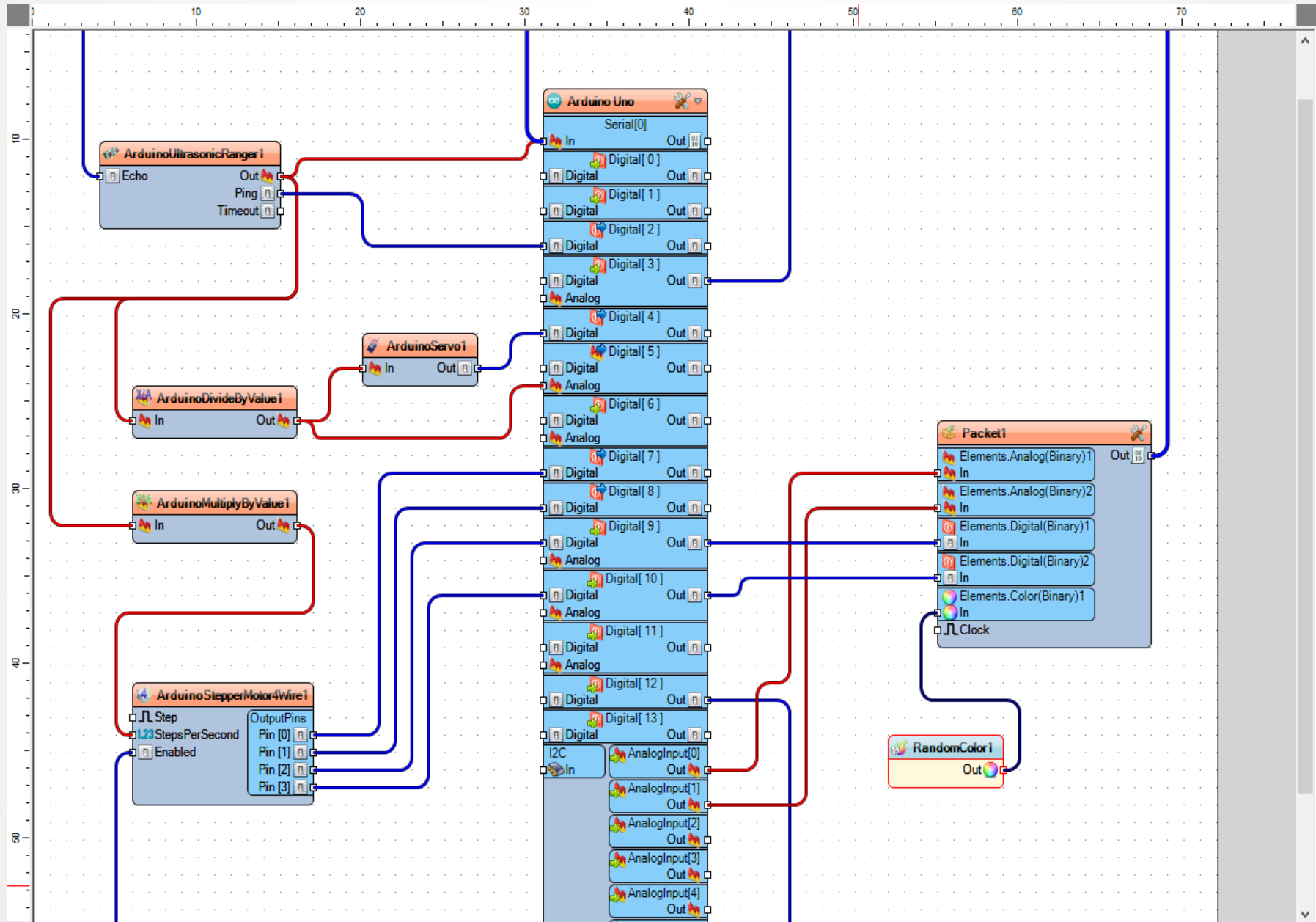


Powerful Object Inspector

The screenshot displays the 'Properties' window in the Arduino IDE, showing the configuration for an Arduino Uno board. The window is organized into a tree view on the left and a detailed view on the right. The 'Digital' section is expanded, showing 14 items (pins 0-13). Each pin has a set of properties: 'AutoConfig' (checked), 'IsOutput', and 'IsPullUp'. The 'Serial' section is also visible, showing one item with properties: 'DataBits' (8), 'Enabled' (checked), 'Parity' (spNone), 'Speed' (9600), and 'StopBits' (1).

Category	Item	Value
Miscellaneous	AnalogInput	6 Items
Miscellaneous	AnalogInputReffere...	Default
Miscellaneous	AnalogOutput	0 Items
Miscellaneous	BoardType	Arduino Uno
Digital	Item [0]	(OutputPin=),Au...
Digital	Item [1]	(OutputPin=),Au...
Digital	Item [2]	(OutputPin=),Au...
Digital	Item [3]	(IsAnalog=False,...
Digital	Item [4]	(OutputPin=),Au...
Digital	Item [4] AutoConfig	<input checked="" type="checkbox"/> True
Digital	Item [4] IsOutput	<input type="checkbox"/> False
Digital	Item [4] IsPullUp	<input type="checkbox"/> False
Digital	Item [5]	(IsAnalog=False,...
Digital	Item [6]	(IsAnalog=False,...
Digital	Item [7]	(OutputPin=),Au...
Digital	Item [7] AutoConfig	<input checked="" type="checkbox"/> True
Digital	Item [7] IsOutput	<input type="checkbox"/> False
Digital	Item [7] IsPullUp	<input type="checkbox"/> False
Digital	Item [8]	(OutputPin=),Au...
Digital	Item [9]	(IsAnalog=False,...
Digital	Item [10]	(IsAnalog=False,...
Digital	Item [11]	(IsAnalog=False,...
Digital	Item [12]	(OutputPin=),Au...
Digital	Item [13]	(OutputPin=),Au...
I2C		(InputPin=())
Serial	Item [0]	300,Parity=spNone
Serial	Item [0] DataBits	8
Serial	Item [0] Enabled	<input checked="" type="checkbox"/> True
Serial	Item [0] Parity	spNone
Serial	Item [0] Speed	9600
Serial	Item [0] StopBits	1
Shields		0 Items

Graphical Designer



Data Visualization

