

# Versalino™ - Prototype to Production Platform

## Atmega328 based Development platform.



### Features

#### Unique Features

- 2 Versalino™ Ports(3 Analog/3 PWM/3 Digital)
- Drop in BT2S Bluetooth to Serial Compatibility
- Versalino™ Programming Port
- Access to source power rail
- Dual Rail capacitive Stabilization by design
- Configurable from minimal and up to save on production and prototyping costs.
- Member of a New Family of Economical and Powerful Versalino™ Controllers

#### Atmega328P\_PU Chip

- Up to 20 MHz Frequency (matched for 16 MHz)
- Low power operation
- 6 Analog, PWM, and digital pins
- one Serial Bus which can be used as digital IOs
- 512 Bytes, EEPROM 32 KB Flash, 2 KB SRAM 1KB

#### Development Features

- Compatible with all Arduino™ Uno & Duemilanove projects.
- Comes with the Arduino™ Bootloader
- Access to the raw power on-board ARM controller through the ICSP port.
- 55mm x 35mm profile

#### Applications

- Robotics Engineering
- Monitor/Control Systems
- Prototyping Platform
- Production & Cost Reduction

#### Industry Standard Configuration

- “Green”/RoHS Device

## DESCRIPTION

The Versalino™ was designed with the process of product development in mind. As such several key changes to the standard Arduino™ Layout were made in an effort to increase the versatility of this new platform.

One of the most critical changes over other platforms available on the market is the fact that the Versalino™ platform capitalizes on the pins already available for use on the Atmega328. These changes allow the use of unstacked upgrade devices (upgrading does not have to make your project taller). Additional changes added functionality for quick uploads from external FTDI programmer and communication through blue tooth functionality.

#### The Standard Bus Advantage:

Though it is possible to program for the Versalino™ using the standard Arduino™ pin-outs. To take advantage of the full power and versatility offered by our new micro-controller platform we have put together a pin-out library that provides you with all of the tools needed to write code that easily takes advantage of the unique Versalino™ bus structure.

The idea is that you may have a device running on USB but need to change it to, or duplicate your codes functionality on BUSA. Instead of having to

change the hard code for every single pin you it is possible to simply change from USB to BUSA in a single location.

The Size:

The Versalino™ Uno Platform and all standard load boards maintain a 55mm by 35mm profile which makes it one of the smaller development platforms without sacrificing any power, or the ability to make field repairs.

This Version of the Versalino™ uses through hole components for all of its interchangeable components which means that using different components does not require that you have surface mount capability, and damaging a part will not force you to replace the entire device.

The use of through hole components is meant to facilitate your design process. A smaller SMD version of the Versalino™ will be available in the near future, and when the time to release you product comes we will be able to support various configurations.

## PIN CONFIGURATION

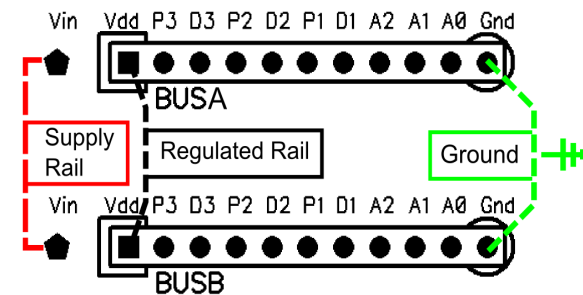
### Versalino™ Primary Buses (BUSA/BUSB):

Great care was taken in the design of the Versalino™ to ensure that an even cross section of the pin types was available to any device that used them. As such each Bus has 3 of each type of Pin Digital, Analog, and PWM.

Additionally the Versalino™ Bus provides access to both regulated and Supply Voltages for any device that uses them. This feature lends toward dramatic design simplification, and allows most designs that would otherwise need to cover an entire board on another platform to take up only a single port.

BUSA	Arduino™ Equiv	Function
Gnd	GND	Ground
A0	A0	Digital/Analog
A1	A1	Digital/Analog
A2	A2	Digital/Analog
D1	2	Digital/Interrupt
P1	3	Digital/PWM
D2	4	Digital
P2	5	Digital/PWM
D3	13	Digital/SCK
P3	6	Digital/PWM
Vdd	5/3.3 Volts[1]	Regulated Power
reserved		Not attached
Vin	Vin	Supply Voltage

[1] Logic Voltage is dependent on supply, regulator, and board configuration changes.

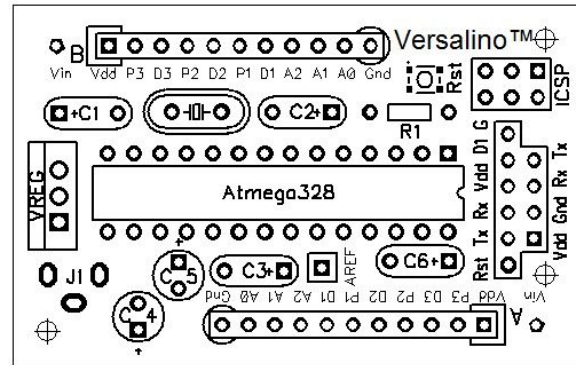


BUSB	Arduino™ Equiv	Function
Gnd	GND	Ground
A0	A3	Digital/Analog
A1	A4	Digital/Analog
A2	A5	Digital/Analog
D1	7	Digital
P1	9	Digital/PWM
D2	8	Digital
P2	10	Digital/PWM
D3	12	Digital/MISO
P3	11	Digital/PWM/MOSI
Vdd	5/3.3 Volts[1]	Regulated Power
reserved		Not attached
Vin	Vin	Supply Voltage

[1] Logic Voltage is dependent on supply, regulator, and board configuration changes.

The Versalino is built using 90 degree female Headers to take advantage of this fact. It is possible to use vertical ports, but loading upgrade boards horizontally makes it possible to support multiple upgrades at once without impacting the height of your

design, or interfering with the functionality of attached devices. It is possible to use cable headers, or solder cables/wires directly to allow for even more flexibility of design.



### Versalino™ COM1 Port:

The COM1 Port is right next to the PGM port, and provides access to the full regulated power rail and the standard Tx and Rx ports. This port is ideally suited for providing a communication link between master and slave units. It also allows units to easily share a single regulated power bus between multiple devices.

COM1	Arduino™ Equiv	Function
Tx	Tx->1	Digital/Serial Tx
Rx	Rx->0	Digital/Serial Rx
Gnd	GND	Ground
Vdd	5/3.3 Volts[1]	Regulated Power

### Versalino™ PGM Port:

The PGM port serves several purposes. The first of which is to provide an easy way to program your chip during development and production. The second function is the ability to easily add a communication link between the Versalino™ and a variety of other Serial devices to include additional Versalino™/Arduino™ platforms. The third function is to simplify the process of managing Firmware Upgrades for complex devices (that use more than one micro-controller).

PGM	Arduino™ Equiv	Function
G	GND	Ground
D1	2	Digital/Interrupt
Vdd	5V	Regulated Voltage
Rx	Rx->0	Digital/Serial Rx
Tx	Tx->1	Digital/Serial Tx
Rst	RESET	Reset pin

