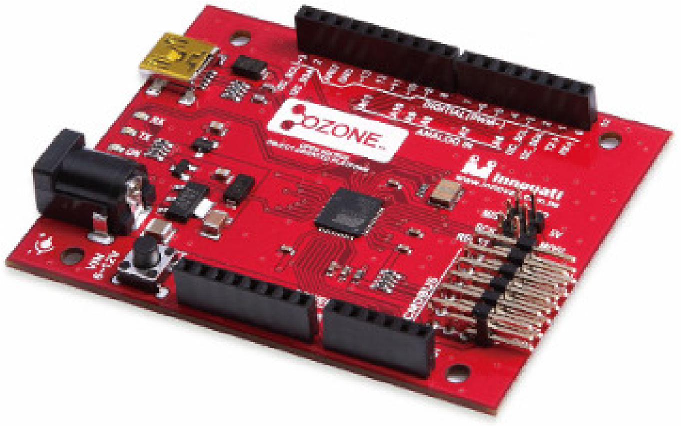
Ozone™

User's Guide

Revision 1.0

Oct. 1, 2013



Passion for innovation

Trademark

Due to continual product improvements, Innovati reserves the right to make modifications to its products without prior notice. Innovati does not recommend the use of its products for application that may present a risk to human life due to malfunction or otherwise.

Disclaimer

Full responsibility for any applications using Innovati products rests firmly with the user and as such Innovati will not be held responsible for any damages that may occur when using Innovati products. This includes damage to equipment or property, personal damage to life or health, damage caused by loss of profits, goodwill or otherwise. Innovati products should not be used for any life saving applications as Innovati’s products are designed for experimental or prototyping purposes only. Innovati is not responsible for any safety, communication or other related regulations. It is advised that children under the age of 14 should only conduct experiments under parental or adult supervision.

Errata

We hope that our users will find this user’s guide a useful, easy to use and interesting publication, as our efforts to do this have been considerable. Additionally, a substantial amount of effort has been put into this user’s guide to ensure accuracy and complete and error free content, however it is almost inevitable that certain errors may have remained undetected. As Innovati will continue to improve the accuracy of its user’s guide, any detected errors will be published on its website. If you find any errors in the user’s guide, please contact us via email service@innovati.com.tw. For the most up-to-date information, please visit our web site at http://www.innovati.com.tw.

Table of Contents

Overview

Innovati’s Ozone™ is a microcontroller board based on Atmel’s ATmega32u4 8-bit microcontroller. The system clock of the Ozone™ board is 16 MHz. The board has 20 digital I/O pins, among which 12 pins can be configured as analog input pins and 7 pins can be configured as PWM (Pulse Width Modulation) output pins.

In addition to the microcontroller’s features, the Ozone™ board provides three Innovati’s CmdBUS™ connectors, which allow users to integrate it with Innovati’s Object-Oriented Modules for more advanced and complicated applications.

Note that this manual mainly describes the functionality of the Ozone™ board. For details of the IDE (Integrated Development Environment) or the C language usage, please refer to their relevant documents.

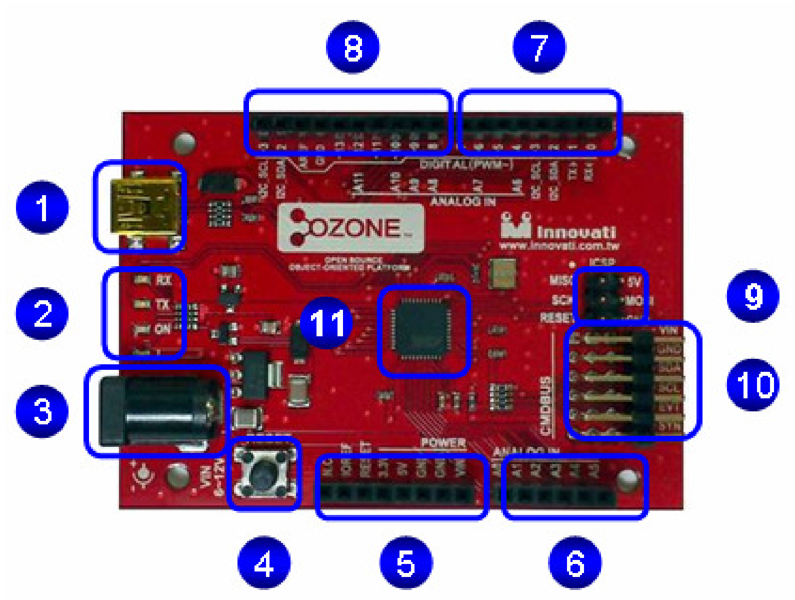
Applications

- Entry-level microcontroller learning kit
- Small scientific project controller
- Interactive art design controller
- Evaluation prototyping
- Integrating with Object-Oriented Modules for sophisticated applications

Features

- Input power supply voltage 7V~12V
- Twenty digital I/O pins
- Seven PWM (Pulse Width Modulation) channels
- Twelve analog input channels (10 bits of resolution)
- 32K bytes Flash Memory of which 4K Bytes used by bootloader
- 2.5K Bytes SRAM
- 1K Bytes EEPROM
- 16 MHz Clock Speed @ 5V
- Module Dimensions: 57.5 mm x 58.6 mm

Specifications



Item	Description
1	Mini USB connector: via a USB cable connecting to computer for downloading and debugging programs.
2	Four LED Indicators. Labeled with RX, TX, ON and L indicating UART data receiving, transmitting, Power On and USB communication.
3	Power Jack for DC 6~12V Power Input.
4	RESET Button. The software resets the Ozone™ on-board microcontroller automatically. If it is not happened, you can start the bootloader by pressing the RESET button. When pressed, the microcontroller will be reset and the USB serial connection will be broken. And then the bootloader will establish the USB connection and execute user's sketch program. The reset procedure requires about 8 seconds.
5	8-pin female connector labeled: N.C.: not connected electrically IOREF: I/O voltage reference (connected to 5V) RESET: connected to RESET button 3.3V : on-board 3.3V output (300mA) 5V: on-board 5V output (800mA) GND: ground Vin: 6~12V power input

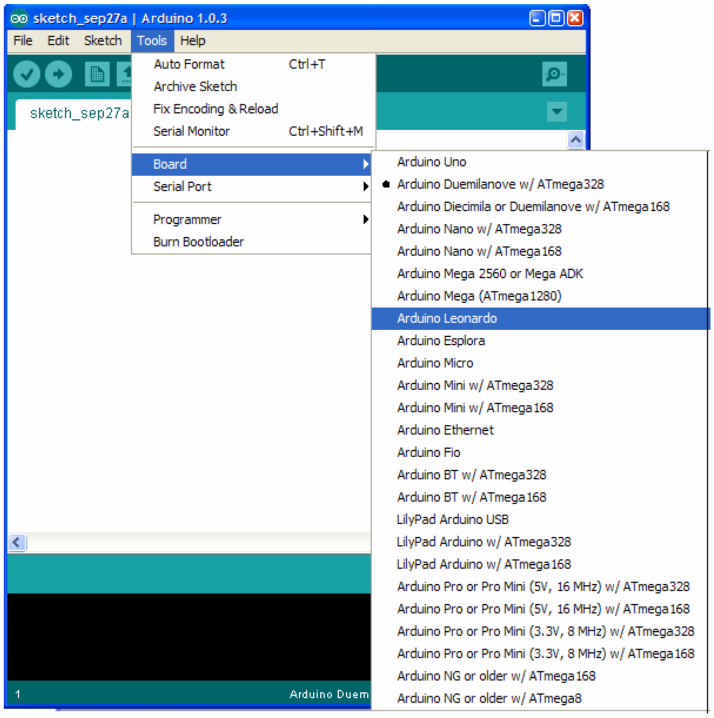
6	6-pin female connector labeled: A0: analog input pin 0 A1: analog input pin 1 A2: analog input pin 2 A3: analog input pin 3 A4: analog input pin 4 A5: analog input pin 5
7	8-pin female connector labeled: 7: digital I/O pin 7 ~6: digital I/O pin 6, PWM output and analog input pin 7 ~5: digital I/O pin 5, PWM output 4: digital I/O pin 4, and analog input pin 6 ~3: digital I/O pin 3, PWM output 2: digital I/O pin 2 1 TX: digital I/O pin 1, UART transmitter 0 RX: digital I/O pin 0, UART receiver
8	10-pin female connector labeled: ~3: digital I/O pin 3 and I2C SCL pin 2: digital I/O pin 2 and I2C SDA pin AREF: analog input reference GND: ground ~13: digital I/O pin 13, PWM output 12: digital I/O pin 12 and analog input pin 11 ~11: digital I/O pin 11, PWM output ~10: digital I/O pin 10, PWM output and analog input pin 10 ~9: digital I/O pin 9, PWM output and analog input pin 9 8: digital I/O pin 8 and analog input pin 8
9	ICSP header: The pins of ICSP header support SPI (Serial peripheral Interface) communication using the SPI library. MOSI, MISO, SCK, RESET, GND, 5V.
10	Three cmdBUS™ connectors for Innovati's Object Oriented Modules (OOMs) connection. The connector is labeled with VIN, GND, SDA, SCL, EVT and SYN. Check the label on board when connecting the cmdBUS™ cable, incorrect insertion may damage the modules.
11	ATmega32U4 microcontroller. 16MHz clock speed; 32K bytes FLASH program memory (4K bytes used by bootloader); 2.5K bytes data SRAM; 1K bytes data EERPOM.

Installing the Development Software

Ozone is using the same Sketch IDE as the Arduino boards. You can download the software from Arduino website http://arduino.cc/en/Mian/Software. Follow the instructions on the website to install the IDE based on the operating system you have.

Running the IDE

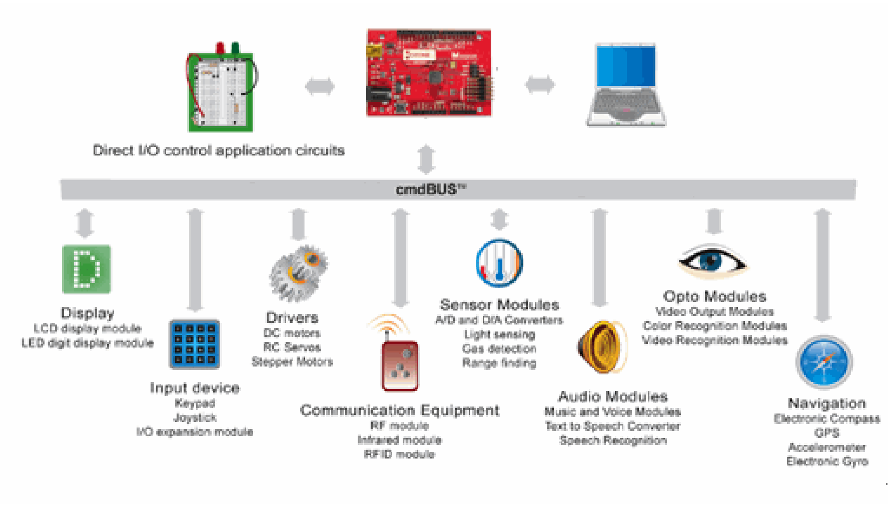
Once the IDE is installed, run the arduino.exe program to open the IDE main (Sketch) window. Then select the Tools menu, Board and then Arduino Leonardo.


sketch_sep27a | Arduino 1.0.3
File
Edit
Sketch
Tools
Help
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Serial Monitor Ctrl+Shift+M
Board
Serial Port
Programmer
Burn Bootloader
sketch_sep27a
Arduino Uno
● Arduino Duemilanove w/ ATmega328
Arduino Diecimila or Duemilanove w/ ATmega168
Arduino Nano w/ ATmega328
Arduino Nano w/ ATmega168
Arduino Mega 2560 or Mega ADK
Arduino Mega (ATmega1280)
Arduino Leonardo
Arduino Esplora
Arduino Micro
Arduino Mini w/ ATmega328
Arduino Mini w/ ATmega168
Arduino Ethernet
Arduino Fio
Arduino BT w/ ATmega328
Arduino BT w/ ATmega168
LilyPad Arduino USB
LilyPad Arduino w/ ATmega328
LilyPad Arduino w/ ATmega168
Arduino Pro or Pro Mini (5V, 16 MHz) w/ ATmega328
Arduino Pro or Pro Mini (5V, 16 MHz) w/ ATmega168
Arduino Pro or Pro Mini (3.3V, 8 MHz) w/ ATmega328
Arduino Pro or Pro Mini (3.3V, 8 MHz) w/ ATmega168
Arduino NG or older w/ ATmega168
Arduino NG or older w/ ATmega8
1
Arduino Duem

If you have problem compiling or downloading the programs to the Ozone™ board, you might need to install the drivers from the Arduino folder.

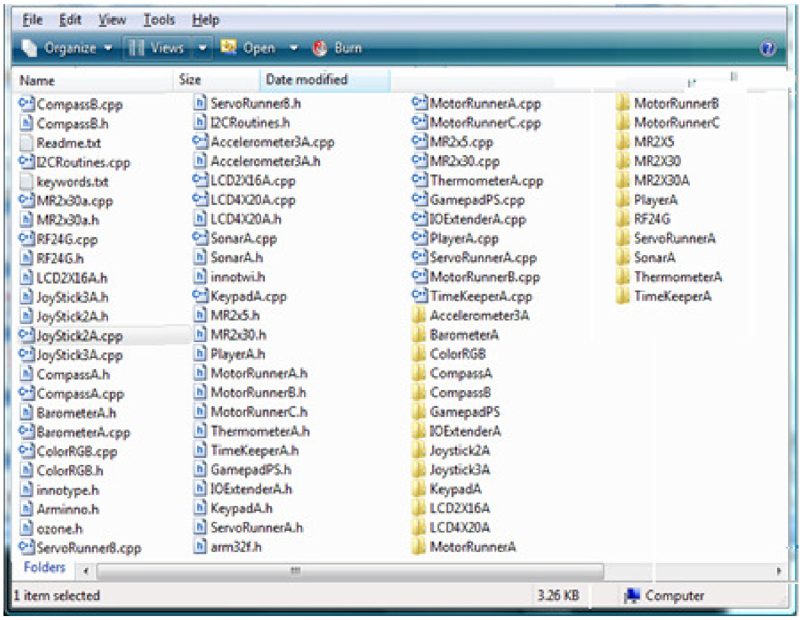
System Scheme

The Ozone™ board supports the conventional direct I/O control for SHIELD applications. However, the most important feature of the Ozone™ board is the cmdBUS™ scheme, which allows up to 32 Smart Modules to be connected to the board.



Using the Object Oriented Modules

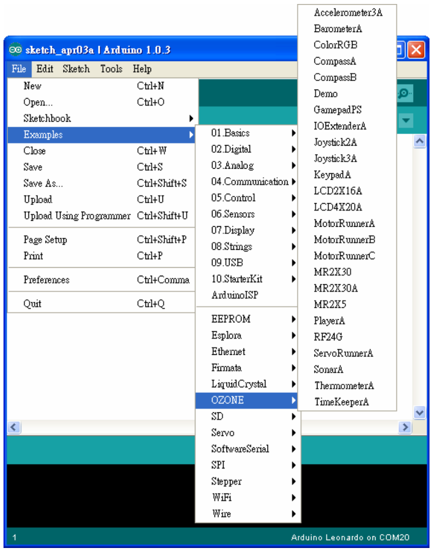
To use the Object Oriented Smart Modules with Ozone™ board, download the Ozone™ Library (Ozone.zip) from our website. Unzip the file and copy the folder to the “libraries” folder where IDE is installed, i.e. c:\arduino\libraries\ozone.



CompassB.cpp
CompassB.h
Readme.txt
I2CRoutines.cpp
keywords.txt
MR2x30a.cpp
MR2x30a.h
RF24G.cpp
RF24G.h
LCD2X16A.h
JoyStick3A.h
JoyStick2A.h
JoyStick2A.cpp
JoyStick3A.cpp
CompassA.h
CompassA.cpp
BarometerA.h
BarometerA.cpp
ColorRGB.cpp
ColorRGB.h
innotype.h
Arminno.h
ozone.h
ServoRunnerB.cpp
ServoRunnerB.h
I2CRoutines.h
Accelerometer3A.cpp
Accelerometer3A.h
LCD2X16A.cpp
LCD4X20A.cpp
LCD4X20A.h
SonarA.cpp
SonarA.h
innotwi.h
KeypadA.cpp
MR2x5.h
MR2x30.h
PlayerA.h
MotorRunnerA.h
MotorRunnerB.h
MotorRunnerC.h
ThermometerA.h
TimeKeeperA.h
GamepadPS.h
IOExtenderA.h
KeypadA.h
ServoRunnerA.h
arm32f.h
MotorRunnerA.cpp
MotorRunnerC.cpp
MR2x5.cpp
MR2x30.cpp
ThermometerA.cpp
GamepadPS.cpp
IOExtenderA.cpp
PlayerA.cpp
ServoRunnerA.cpp
MotorRunnerB.cpp
TimeKeeperA.cpp
Accelerometer3A
BarometerA
ColorRGB
CompassA
CompassB
GamepadPS
IOExtenderA
Joystick2A
Joystick3A
KeypadA
LCD2X16A
LCD4X20A
MotorRunnerA
MotorRunnerB
MotorRunnerC
MR2X5
MR2X30
MR2X30A
PlayerA
RF24G
ServoRunnerA
SonarA
ThermometerA
TimeKeeperA

1 item selected 3.26 KB Computer

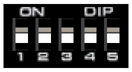
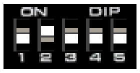
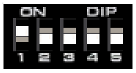
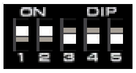
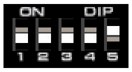
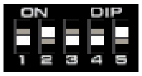
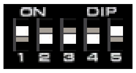
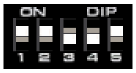
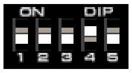
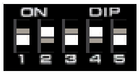
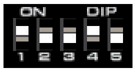
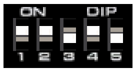
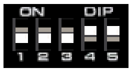
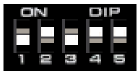
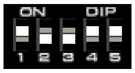
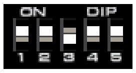
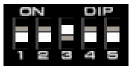
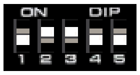
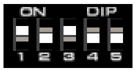
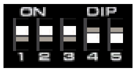
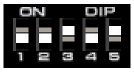
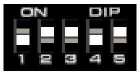
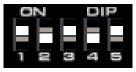
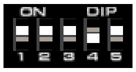
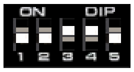
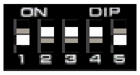
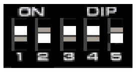
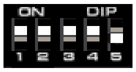
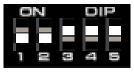
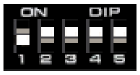
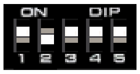
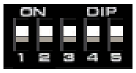
Once the library folder is copied, open the IDE and you can find an Ozone submenu in the Examples menu item with all the smart module example programs. Select the program that you want to open.



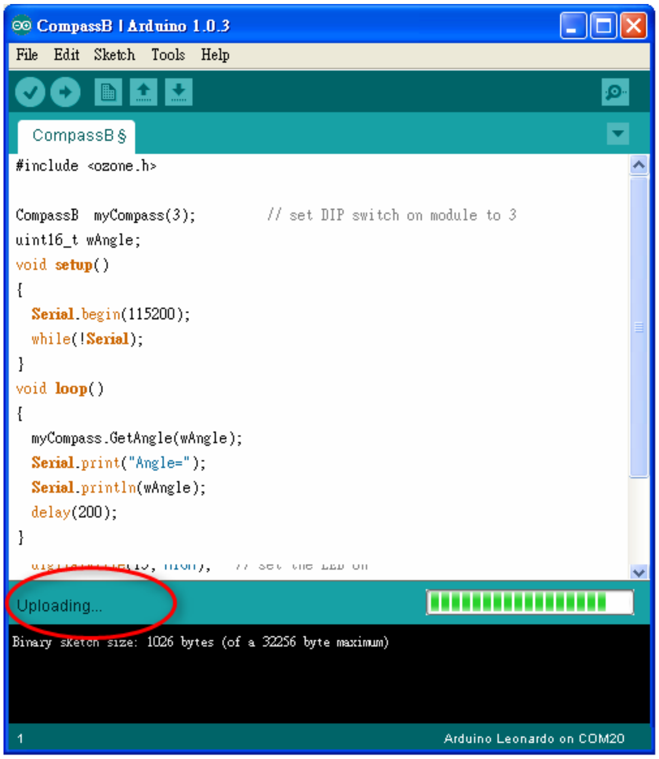
After the program is loaded, you can see the code in the editing window. Compile and upload to the Ozone board. Some example programs can be found in the Appendix section.

Connecting the Object Oriented Modules

Connect the cmdBUS™ cable of the Smart Module to the cmdBUS™ pins on the Ozone™ board. Set the module's DIP switch to corresponding ID set in the program.

0		**8**		**16**		**24**	
1		**9**		**17**		**25**	
2		**10**		**18**		**26**	
3		**11**		**19**		**27**	
4		**12**		**20**		**28**	
5		**13**		**21**		**29**	
6		**14**		**22**		**30**	
7		**15**		**23**		**31**	

After the module is connected to the Ozone™ board, press the uploading button for compiling and loading program code to the board, you are ready to explore the smart module features.



```
#include <ozone.h>

CompassB  myCompass(3);        // set DIP switch on module to 3
uint16_t wAngle;
void setup()
{
  Serial.begin(115200);
  while(!Serial);
}
void loop()
{
  myCompass.GetAngle(wAngle);
  Serial.print("Angle=");
  Serial.println(wAngle);
  delay(200);
}
  digitalWrite(13, HIGH);    // set the LED on
```

```
Binary sketch size: 1026 bytes (of a 32256 byte maximum)
```

For more information on the programming syntax the Smart Modules, download the latest version of the corresponding manual from our website.

Appendix --- Tutorial Programs

To help you be familiar with the Ozone^{TM} featured Object Oriented Module (OOM) usage, several example programs are provided here. This section is given to introduce the unique features of some of the modules. However, users who are not familiar with C language and/or the Arduino function basics, refer to their relevant documents.

To maintain the tutorial programs free of error and up-to-date, they are subject to change without notice.

Ex. 1 --- Using the 2X16A LCD Module

```
#include <ozone.h>

LCD2X16A myLCD(0);                  // declare Module ID as 0

void setup()
{
   myLCD.BacklightOn(0);            // turn on the LCD backlight
   myLCD.Display("Hello");          // display “Hello”
}

void loop()
{
}
```

Ex. 2 --- Using the Keypad Module

```
#include <ozone.h>

KeypadA myKey(1);                   // declare Module ID as 1
uint8_t Status, KeyID;

void setup()
{
   myKey.SetKeypadMode(2);               // set keypad to mode 2 (number mode)
}
```

```
void loop()
{
   Status = myKey.GetKeyID(KeyID);    // polling keypad input
   if(Status) Serial. print(KeyID);          // display in serial monitor, if key pressed
   delay(200);                                        // pause 200ms
}
```

Ex. 3 --- Using the Sonar Module

```
#include <ozone.h>
SonarA mySonar(16);              // declare Module ID as 16

uint8_t Status;
uint16_t Distance;

void setup()
{
}
void loop()
{
    do {
        mySonar.Ranging()  ;               // range finding
        delay(100);                            // pause 100ms
        Status= mySonar.GetDistance(1, Distance);    // get range finding result
    }
    while(Status != 1);                      // wait for valid result
    Serial.print("Distance=");            // display text in serial monitor
    Serial.println(Distance);               // display result in serial monitor
}
```

Ex. 4 --- Using the Compass Module

```
#include <ozone.h>

CompassB    myCompass(3);          // declare Module ID as 3

uint16_t wAngle;
```

```
void setup()
{
}
void loop()
{
    myCompass.GetAngle(wAngle);
    Serial.print("Angle=");                  // display text in serial monitor
    Serial.println(wAngle);                  // display result in serial monitor
    delay(200);                              // pause 200ms
}
```