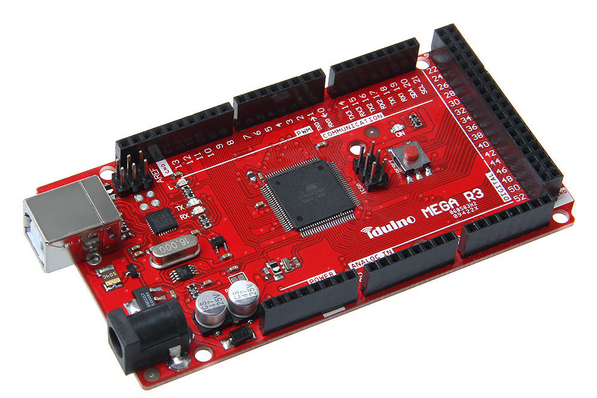
20150205 Geeetech Iduino Mega R3 Board Microcontroller ATmega2560 ATMEGA16U2 for Arduino

[Reference](http://www.geeetech.com/wiki/index.php/Iduino_MEGA_R3)

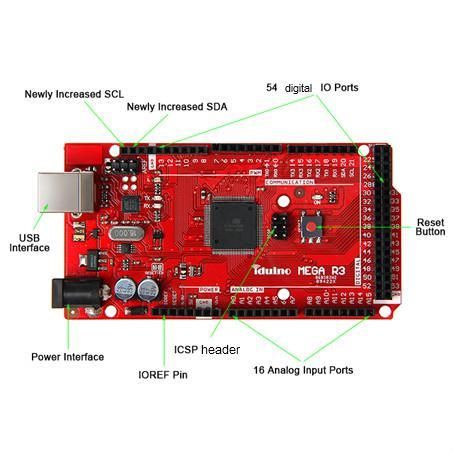
Will try to load LeRoy’s program for the Hero Jr Robot for testing purposes.



## Description

As with our Iduino Mega 2560, the Iduino MEGA R3 is also a microcontroller board based on the ATmega2560. It has 54 digital input/output pins (of which 14 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, an external power jack, an ICSP header, and a reset button, enabling the board to be plug-and-play.

The Iduino MEGA R3 differs from Iduino Mega 2560 in that it has 3 additional pins, they are SDA, SCL and IOREF,amoung which the IOREF pin can provide the VREF to the board. In addition, considering the compatibility of Iduino MEGA R3 with the series products to come，Iduino MEGA R3 has a port reserved on the left side of IOREF pin.



**Interfaces specifications**

Microcontroller：ATmega2560  
Input Voltage (recommended): 7-12V  
Operating Voltage： 5V

54 Digital I/O Pins：（of which 14 provide PWM output）

16 Analog Input

4 UART (hardware serial ports)

An USB interface

An ICSP header

A power interface

A reset button

DC Current of I/O Pin: 40 mA

DC Current for 3.3V Pin: 50mA

Flash Memory: 256KB of which 8KB is used for bootloader ( Atmega 328)

SRAM: 8KB

EEPROM： 4 KB

Clock Speed: 16MHz

Product Weight：34g

Dimensions：102mm\*54mm\*11mm

[Please click here to find more information.](http://www.geeetech.com/wiki/index.php/Iduino_MEGA_R3)

**Package list:**

1 x Geeetech Iduino Mega R3

## The Interface Resource Instruction

**Power Interface**: 7-12 V，supply power for the entire board, note that higher voltage may cause damage to the chip.

**54 digital IO ports**: use pinMode (), digitalWrite () and digitalRead () function to manipulate, each IO can be used as input and output ports, the operating voltage is 5V, the output of each channel can provide or receive a maximum current of 40mA and Each channel is configured with a 20-50K ohm internal pull-up resistor (disconnected by default).

In addition, some pins have the following specific functions:

**4 serial ports**：Serial port 0---0(RX) and 1(TX); serial port 1---19(RX) and 18(TX); serial port 2---17(RX) and 16(TX); serial port 3---15(RX) and 14(TX). Pins 0 and 1are connected to the corresponding pins of the ATmega16U2 USB-to-TTL Serial chip, used to receive (RX) and transmit (TX) TTL serial data.

**6 external interrupts**: 2 (interrupt 0), 3 (interrupt 1), 18 (interrupt 5), 19 (interrupt 4), 20 (interrupt 3) and 21 (Interrupt 2). These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value.

**14 PWM** (0--13): Provide 14 8-bit PWM output with the analogWrite() function.

**SPI** (53 (SS), 51 (MOSI), 50 (MISO), 52 (SCK)): support SPI communication.

**LED**（pin 13）：a pin specially reserved for Iduino to test the LED When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.

**ICSP header**：The ICSP header is composed of 6 pins. Usually used to upload Bootloader（You can complete the connection between PC and board with our product USB Tiny OTP Writer）

**16 Analog input ports**：Each of which can convert the analog input into 10 bits of resolution digital input (i.e. 0-1023 different values). By default they measure from ground to 5 volts, but it is possible to change the upper limit of their range using the AREF pin and analogReference() function.

Besides, some pins have other specific functions： TWI interfaces (20 (SDA) and 21 (SCL); new SDA and SCL): Support communications interface (compatible with I2C bus).

**USB Interface**: This interface is used to load the main program or connected with external USB devices.

**IOREF**: This pin is used to provide the operating reference voltage of the microcontroller on Iduino.

# Development Environment setting

## Interface Connecting and Setting

If you want to load or test the program of Iduino Mega R3, just connect the Iduino Mega R3 with computer with a USB cable, as shown below：

ORIGINAL CODE

pin#include <Wire.h>

#include <Keypad.h>

#include <NewPing.h>

/\*Hero Jr Control - Arduino Mega

Commands so far:

key() - read a button push from the keypad

ping() - read distance from ping sensor

led(#, state) - turn on led # 0 to 7 state is on or off

forward(distance, speed) - move forward so far at a speed

backward(distance, speed) - move backward so far at a speed

left(degree) - turn to the left

right(degree) - turn to the right

light() - read the light sensor

volt() - check the voltage levels for the drive battery

test() - don't know yet

jrinit() - don't know yet

\*\*\* Voltage Sensor Information

This seems to work 3.3v hooked to output side

and "S" hooked to Analog input 1

It qives whole numbers as output

Tested on the UNO, and not sure why it needs "Wire.h"

\*/

int lightPin = 0; //analog pin0 for photoresistor

#define TRIGGER\_PIN 13 // Arduino pin tied to trigger pin on the ultrasonic sensor.

#define ECHO\_PIN 12 // Arduino pin tied to echo pin on the ultrasonic sensor.

#define MAX\_DISTANCE 200 // Maximum distance we want to ping for (in centimeters). Maximum sensor distance is rated at 400-500cm. About 78 Inches

NewPing sonar(TRIGGER\_PIN, ECHO\_PIN, MAX\_DISTANCE); // NewPing setup of pins and maximum distance.

int val11;

int val2;

int led0 = 22;

int led1 = 24;

int led2 = 26;

int led3 = 28;

int led4 = 30;

int led5 = 32;

int led6 = 34;

int led7 = 36;

int state0 = 1;

int state1 = 1;

int state2 = 1;

int state3 = 1;

int state4 = 1;

int state5 = 1;

int state6 = 1;

int state7 = 1;

const byte ROWS = 4; // Four rows

const byte COLS = 4; // four columns

// Define the Keymap

char keys[ROWS][COLS] = {

{'C','D','E','F'},

{'8','9','A','B'},

{'4','5','6','7'},

{'0','1','2','3'}

};

// Connect keypad ROW0, ROW1, ROW2 and ROW3 to these Arduino pins.

byte rowPins[ROWS] = { 4, 5, 6, 7 };

// Connect keypad COL0, COL1 and COL2 to these Arduino pins.

byte colPins[COLS] = { 8, 9, 10, 11 };

// Create the Keypad

Keypad kpd = Keypad( makeKeymap(keys), rowPins, colPins, ROWS, COLS );

int number;

int state;

String command;

int distance;

int speed;

int degree;

int fl;

void setup()

{

Serial.begin(9600);

pinMode(led0, OUTPUT);

pinMode(led1, OUTPUT);

pinMode(led2, OUTPUT);

pinMode(led3, OUTPUT);

pinMode(led4, OUTPUT);

pinMode(led5, OUTPUT);

pinMode(led6, OUTPUT);

pinMode(led7, OUTPUT);

digitalWrite(led0, 1);

digitalWrite(led1, 1);

digitalWrite(led2, 1);

digitalWrite(led3, 1);

digitalWrite(led4, 1);

digitalWrite(led5, 1);

digitalWrite(led6, 1);

digitalWrite(led7, 1);

}

void loop()

{

fl = 0;

if (Serial.available() >0)

{

command = Serial.readStringUntil(';');

if (command == "led") {

number = Serial.parseInt();

led (number);

fl = 1;

}

if (command == "key") { key(); fl = 1; }

if (command == "ping") { ping(); fl = 1; }

if (command == "forward") { distance = Serial.parseInt();

speed = Serial.parseInt();

forward(distance, speed);

fl = 1;

}

if (command == "backward") { distance = Serial.parseInt();

speed = Serial.parseInt();

backward(distance, speed);

fl = 1;

}

if (command == "left") { degree = Serial.parseInt();

left(degree);

fl = 1;

}

if (command == "right") { degree = Serial.parseInt();

right(degree);

fl = 1;

}

if (command == "light") { light(); fl = 1;}

if (command == "volt") { volt(); fl = 1;}

if (command == "test") { test(); fl = 1;}

if (command == "jrinit") { jrinit(); fl = 1;}

if (fl == '0') {

Serial.println("ERR");

fl = 0;

}

}

}

void key() {

delay(100);

char key = kpd.getKey();

Serial.println(key);

}

void ping() {

unsigned int uS = sonar.ping(); // Send ping, get ping time in microseconds (uS).

Serial.println(uS / US\_ROUNDTRIP\_IN); // Convert ping time to distance in cm and print result (0 = outside set distance range)

}

void led(int number1) {

switch (number1) {

case 0:

state0 = !state0;

digitalWrite(led0, state0);

break;

case 1:

state1 = !state1;

digitalWrite(led1, state1);

break;

case 2:

state2 = !state2;

digitalWrite(led2, state2);

break;

case 3:

state3 = !state3;

digitalWrite(led3, state3);

break;

case 4:

state4 = !state4;

digitalWrite(led4, state4);

break;

case 5:

state5 = !state5;

digitalWrite(led5, state5);

break;

case 6:

state6 = !state6;

digitalWrite(led6, state6);

break;

case 7:

state7 = !state7;

digitalWrite(led7, state7);

break;

}

}

void forward(int distance, int speed) {

}

void backward(int distance, int speed) {

}

void left(int degree) {

}

void right(int degree) {

}

void light() {

int photor = analogRead(lightPin);

Serial.println(photor);

delay(50);

}

void volt() {

float temp;

val11=analogRead(1);

temp=val11/4;

val11=(int)temp;

val2=((val11)/10);

Serial.println(val2);

}

void test() {

}

void jrinit() {

}