

TENDA N301

New Design More Easily

Wireless N300 Easy Setup Router N301

The N301 Wireless N300 Easy Setup Router is designed to setup more easily for the home user. It complies with IEEE802.11n, delivers wireless speeds of up to 300Mbps, making it perfect for everyday web activities like email, chat, streaming videos, online gaming and more. The N301 can also work as a client router to connect ISP network wirelessly or uplink AP to share the Internet to every corner, eliminating the dead point.



Easy to Setup

With just three easy steps, get your secure wireless network up and running in minutes.



Secure WiFi at a push of WPS button

Compatible with WI-FI Protected Setup™ (WPS), N301 features WPS that allows users to almost instantly setup their security simply by pressing the "WPS" button automatically establishing a secure connection. Not only is this faster than normal security setups but more convenient in that you don't need to remember a password.



Advanced Wireless Security

N301 offers multi-level wireless encryption options to prevent unauthorized access and protect your important data.

1. 64/128bit WEP、WPA-PSK、WPA2-PSK

2. Wireless Access Control based on the mac address of wireless adapter

Wireless WAN

N301 can work as client mode to connect to ISP network or uplink AP wirelessly to share the Internet to every corner, eliminating the dead point.





Parental Control

Support parental control function like IP/MAC filter, you can limit the time to access the internet and block the websites.



How it Works

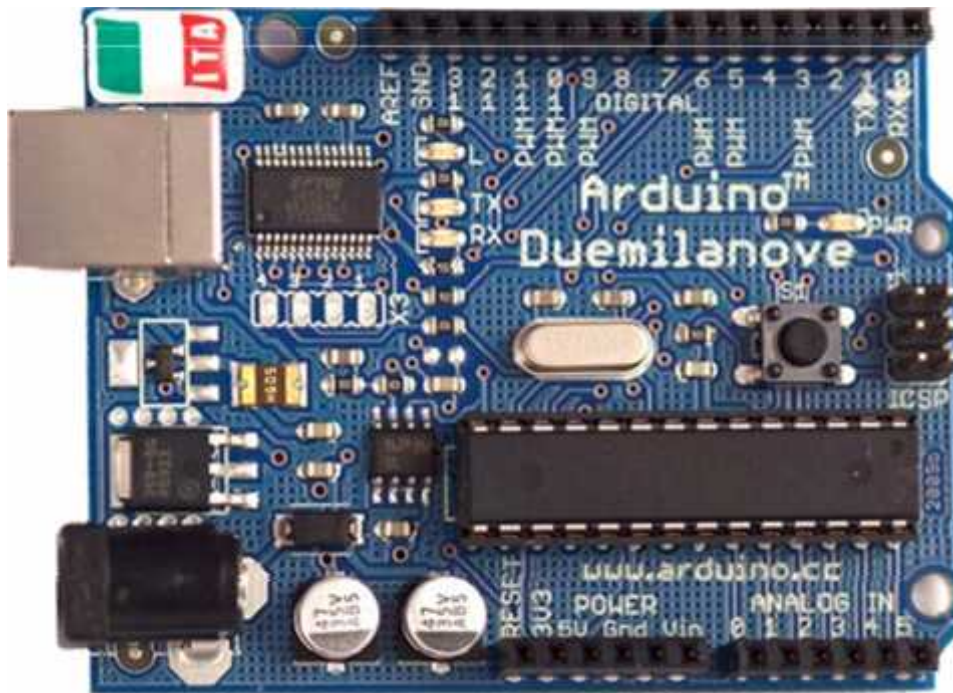
How to make wireless magic, start with Internet service and a modem then connect a Linksys router. The router broadcasts a wireless signal that lets you easily connect your devices and smart appliances.

Specifications

Hardware Features	
Compliant Standard	IEEE 802.3/3U IEEE 802.11n/g/b
Wired Interface	1 10/100Mbps WAN Port 3 10/100Mbps LAN Ports
Antennas	2 fixed 5dbi Omni Directional antennas
Button	1 Reset/WPS Button
Item dimensions(L*W*H)	127.4mm*90.5mm*26mm
Wireless Features	
Frequency Range (GHz)	2.412GHz-2.472GHz
Wireless Link Rate	IEEE 802.11n: up to 300Mbps IEEE 802.11g: up to 54Mbps IEEE 802.11b: up to 11Mbps
Wireless Security	64/128bit WEP WPA-PSK WPA2-PSK WPS support
Working Mode	AP WISP WDS Bridge
Wireless Function	Enable/Disable Wireless Radio Wireless Access Control
Reception Sensitivity	b mode : 1M -92dBm@8% 11M -87dBm@8% PER ; g mode : 54M -72dBm@8% PER ; n mode : 72.2M -68dBm@8% PER ; 150M -68dBm@8% PER ;

Software Features	
Internet Connection Type	Dynamic IP、PPPOE、Static IP、L2TP、PPTP
DHCP	Built-in DHCP server DHCP Client List Address Reservation
Virtual Server	Port Forwarding DMZ Host
Parental Control	Client Filter Mac Filter Websites Filter
Dynamic DNS	No-IP dyndns
VPN Pass-Through	PPTP L2TP
Other Advanced Function	Bandwidth Control Mac Address Clone Remote Web Management System Log
Others	
Package Contents	Wireless N300 Easy Setup Router N301 Quick Installation Guide Ethernet cable Power adapter Resources CD
Environment	Operating Temperature: 0℃~40℃ Storage Temperature: -40℃~70℃ Operating Humidity: 10%~90% non-condensing Storage Humidity: 5%~90% non-condensing
Certificates	FCC, CE, RoHs

ARDUINO DUEMILANOVE



Product Overview

The Arduino Duemilanove ("2009") is a microcontroller board based on the ATmega328. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.

"Duemilanove" means 2009 in Italian and is named after the year of its release. The Duemilanove is the latest in a series of USB Arduino boards; for a comparison with previous versions, see the [index of Arduino boards](#).

Index

Technical Specifications

Page 2

How to use Arduino
Programming Environment, Basic Tutorials

Page 6

Terms & Conditions

Page 7

Environmental Policies
half sqm of green via Impatto Zero®

Page 7



radiospares

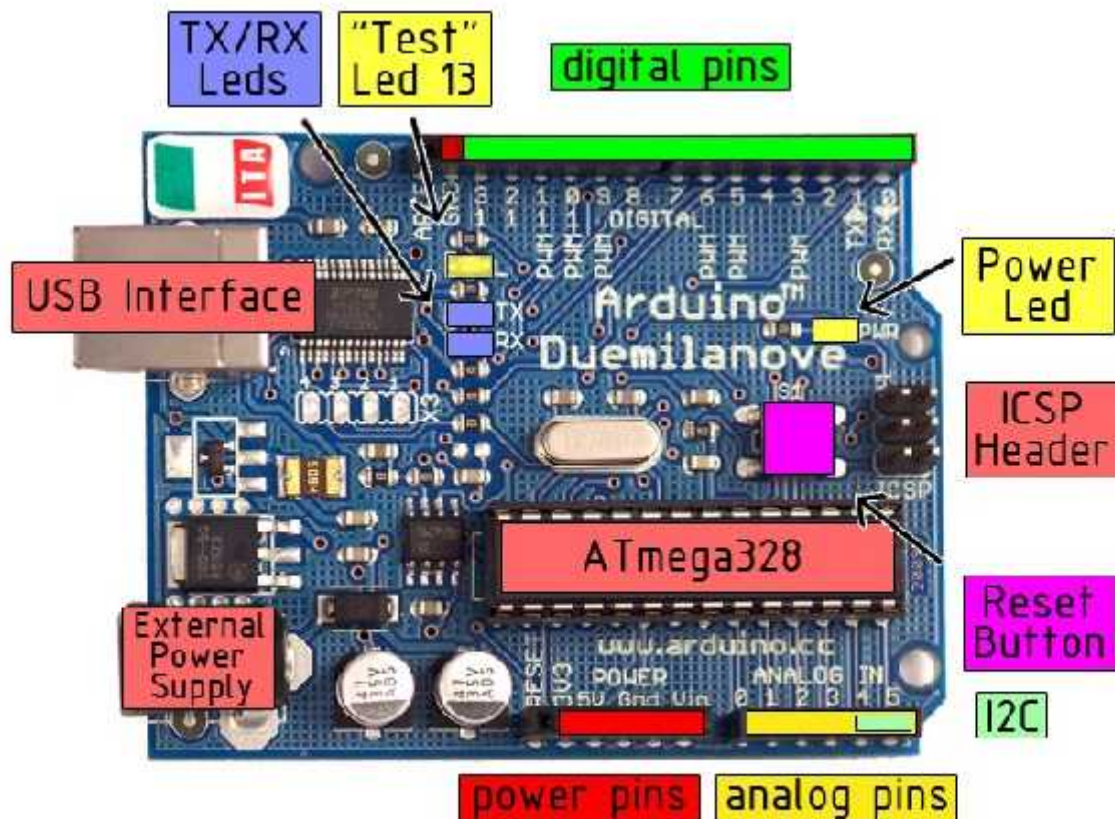
RADIONICS



Summary

Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB of which 2 KB used by bootloader
SRAM	2 KB
EEPROM	1 KB
Clock Speed	16 MHz

the board



Power

The Arduino Duemilanove can be powered via the USB connection or with an external power supply. The power source is selected automatically.

External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector.

The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

The power pins are as follows:

- **VIN.** The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- **5V.** The regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply.
- **3V3.** A 3.3 volt supply generated by the on-board FTDI chip. Maximum current draw is 50 mA.
- **GND.** Ground pins.

Memory

The Atmega328 has 32 KB of flash memory for storing code (of which 2 KB is used for the bootloader); the ATmega328 has 32 KB, (also with 2 KB used for the bootloader). The Atmega328 has 2 KB of SRAM and 1 KB of EEPROM (which can be read and written with the [EEPROM library](#)).

Input and Output

Each of the 14 digital pins on the Duemilanove can be used as an input or output, using [pinMode\(\)](#), [digitalWrite\(\)](#), and [digitalRead\(\)](#) functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

- **Serial: 0 (RX) and 1 (TX).** Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the FTDI USB-to-TTL Serial chip.
- **External Interrupts: 2 and 3.** These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the [attachInterrupt\(\)](#) function for details.
- **PWM: 3, 5, 6, 9, 10, and 11.** Provide 8-bit PWM output with the [analogWrite\(\)](#) function.
- **SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK).** These pins support SPI communication, which, although provided by the underlying hardware, is not currently included in the Arduino language.
- **LED: 13.** There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.



RADIOSPARES

RADIONICS



The Duemilanove has 6 analog inputs, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the AREF pin and the [analogReference\(\)](#) function. Additionally, some pins have specialized functionality:

- **I²C: 4 (SDA) and 5 (SCL).** Support I²C (TWI) communication using the [Wire library](#).

There are a couple of other pins on the board:

- **AREF.** Reference voltage for the analog inputs. Used with [analogReference\(\)](#).
- **Reset.** Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

See also the [mapping between Arduino pins and Atmega328 ports](#).

Communication

The Arduino Duemilanove has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega328 provide UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An FTDI FT232RL on the board channels this serial communication over USB and the [FTDI drivers](#) (included with the Arduino software) provide a virtual com port to software on the computer. The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board. The RX and TX LEDs on the board will flash when data is being transmitted via the FTDI chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

A [SoftwareSerial library](#) allows for serial communication on any of the Duemilanove's digital pins.

The ATmega328 also support I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus; see the [documentation](#) for details. To use the SPI communication, please see the ATmega328 datasheet.

Programming

The Arduino Duemilanove can be programmed with the Arduino software ([download](#)). Select "Arduino Duemilanove w/ ATmega328" from the **Tools > Board** menu (according to the microcontroller on your board). For details, see the [reference](#) and [tutorials](#).

The ATmega328 on the Arduino Duemilanove comes preburned with a [bootloader](#) that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol ([reference](#), [C header files](#)).

You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header; see [these instructions](#) for details.



radiospares

RADIONICS



Automatic (Software) Reset

Rather than requiring a physical press of the reset button before an upload, the Arduino Duemilanove is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the FT232RL is connected to the reset line of the ATmega328 via a 100 nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the bootloader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload.

This setup has other implications. When the Duemilanove is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the Duemilanove. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data.

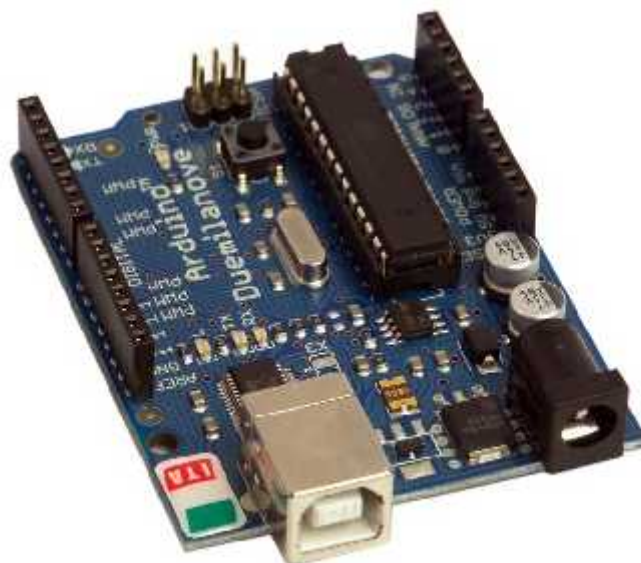
The Duemilanove contains a trace that can be cut to disable the auto-reset. The pads on either side of the trace can be soldered together to re-enable it. It's labeled "RESET-EN". You may also be able to disable the auto-reset by connecting a 110 ohm resistor from 5V to the reset line; see [this forum thread](#) for details.

USB Overcurrent Protection

The Arduino Duemilanove has a resettable polyfuse that protects your computer's USB ports from shorts and overcurrent. Although most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed.

Physical Characteristics

The maximum length and width of the Duemilanove PCB are 2.7 and 2.1 inches respectively, with the USB connector and power jack extending beyond the former dimension. Three screw holes allow the board to be attached to a surface or case. Note that the distance between digital pins 7 and 8 is 160 mil (0.16"), not an even multiple of the 100 mil spacing of the other pins.



RADIOSPARES

RADIONICS





How to use Arduino



Arduino can sense the environment by receiving input from a variety of sensors and can affect its surroundings by controlling lights, motors, and other actuators. The microcontroller on the board is programmed using the [Arduino programming language](#) (based on [Wiring](#)) and the Arduino development environment (based on [Processing](#)). Arduino projects can be stand-alone or they can communicate with software on running on a computer (e.g. Flash, Processing, MaxMSP).

Arduino is a cross-platform program. You'll have to follow different instructions for your personal OS. Check on the [Arduino site](#) for the latest instructions. <http://arduino.cc/en/Guide/HomePage>

Linux Install

Windows Install

Mac Install

Once you have downloaded/unzipped the arduino IDE, you'll need to install the FTDI Drivers to let your PC talk to the board. First **Plug the Arduino to your PC via USB cable.**

Blink led

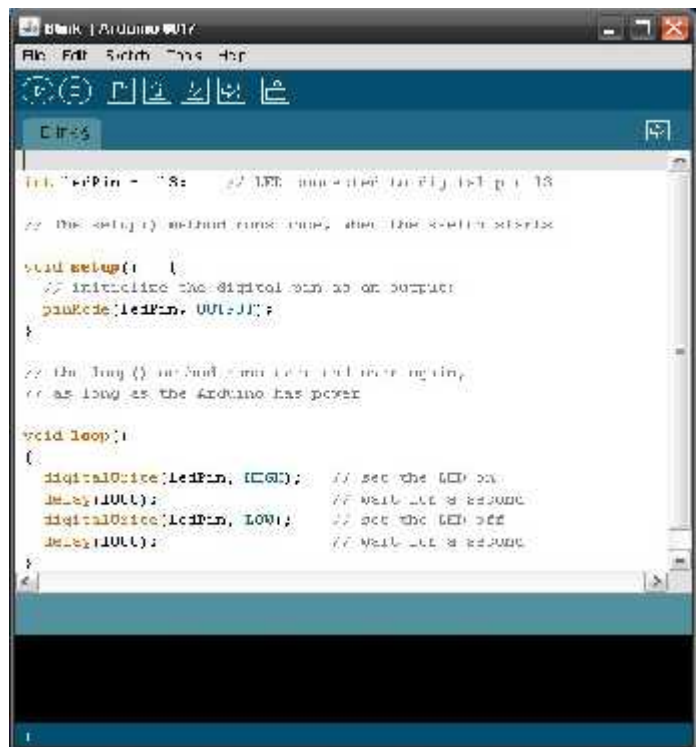
Now you're actually ready to "burn" your first program on the arduino board. To select "blink led", the physical translation of the well known programming "hello world", select

**File>Sketchbook>
Arduino-0017>Examples>
Digital>Blink**

Once you have your sketch you'll see something very close to the screenshot on the right.

In **Tools>Board** select

Now you have to go to **Tools>SerialPort** and select the right serial port, the one arduino is attached to.



Press Compile button (to check for errors)

Upload

TX RX Flashing

Blinking Led!

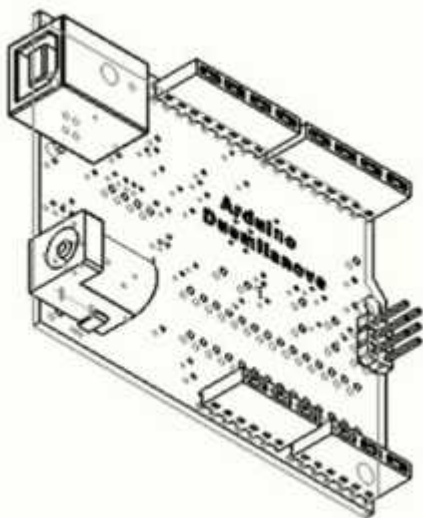
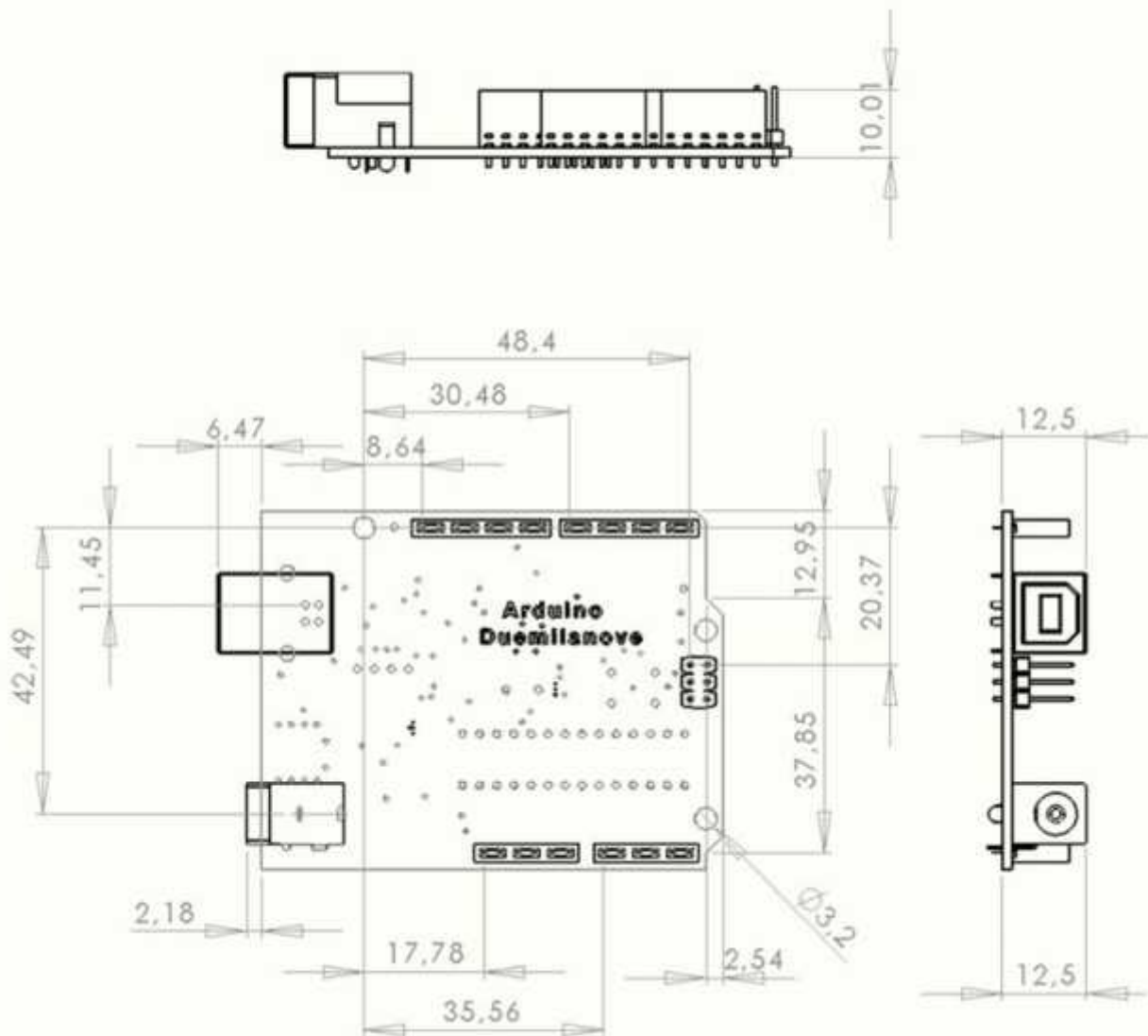


RADIOSPARES

RADIONICS



Dimensioned Drawing



radiospares

RADIONICS





1. Warranties

1.1 The producer warrants that its products will conform to the Specifications. This warranty lasts for one (1) years from the date of the sale. The producer shall not be liable for any defects that are caused by neglect, misuse or mistreatment by the Customer, including improper installation or testing, or for any products that have been altered or modified in any way by a Customer. Moreover, The producer shall not be liable for any defects that result from Customer's design, specifications or instructions for such products. Testing and other quality control techniques are used to the extent the producer deems necessary.

1.2 If any products fail to conform to the warranty set forth above, the producer's sole liability shall be to replace such products. The producer's liability shall be limited to products that are determined by the producer not to conform to such warranty. If the producer elects to replace such products, the producer shall have a reasonable time to replacements. Replaced products shall be warranted for a new full warranty period.

1.3 EXCEPT AS SET FORTH ABOVE, PRODUCTS ARE PROVIDED "AS IS" AND "WITH ALL FAULTS." THE PRODUCER DISCLAIMS ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, REGARDING PRODUCTS, INCLUDING BUT NOT LIMITED TO, ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE

1.4 Customer agrees that prior to using any systems that include the producer products, Customer will test such systems and the functionality of the products as used in such systems. The producer may provide technical, applications or design advice, quality characterization, reliability data or other services. Customer acknowledges and agrees that providing these services shall not expand or otherwise alter the producer's warranties, as set forth above, and no additional obligations or liabilities shall arise from the producer providing such services.

1.5 The Arduino™ products are not authorized for use in safety-critical applications where a failure of the product would reasonably be expected to cause severe personal injury or death. Safety-Critical Applications include, without limitation, life support devices and systems, equipment or systems for the operation of nuclear facilities and weapons systems. Arduino™ products are neither designed nor intended for use in military or aerospace applications or environments and for automotive applications or environment. Customer acknowledges and agrees that any such use of Arduino™ products which is solely at the Customer's risk, and that Customer is solely responsible for compliance with all legal and regulatory requirements in connection with such use.

1.6 Customer acknowledges and agrees that it is solely responsible for compliance with all legal, regulatory and safety-related requirements concerning its products and any use of Arduino™ products in Customer's applications, notwithstanding any applications- related information or support that may be provided by the producer.

2. Indemnification

The Customer acknowledges and agrees to defend, indemnify and hold harmless the producer from and against any and all third-party losses, damages, liabilities and expenses it incurs to the extent directly caused by: (i) an actual breach by a Customer of the representation and warranties made under this terms and conditions or (ii) the gross negligence or willful misconduct by the Customer.

3. Consequential Damages Waiver

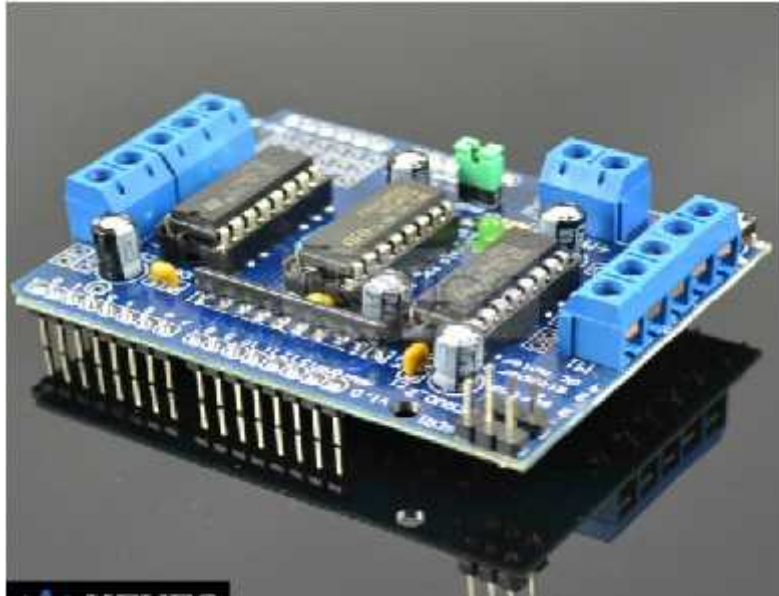
In no event the producer shall be liable to the Customer or any third parties for any special, collateral, indirect, punitive, incidental, consequential or exemplary damages in connection with or arising out of the products provided hereunder, regardless of whether the producer has been advised of the possibility of such damages. This section will survive the termination of the warranty period.

4. Changes to specifications

The producer may make changes to specifications and product descriptions at any time, without notice. The Customer must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." The producer reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The product information on the Web Site or Materials is subject to change without notice. Do not finalize a design with this information.



L293D motor control shield



L293D is a monolithic integrated, high voltage, high current, 4-channel driver. Basically this means using this chip you can drive DC motors with power supplier up to 36 Volts, and the chip can supply a maximum current of 600mA per channel. L293D chip is also known as a type of H-Bridge. The H-Bridge is typically an electrical circuit that enables a voltage to be applied across a load in either direction to an output, e.g. motor.

Features:

- 2 connections for 5V 'hobby' servos connected to the Arduino's high-resolution dedicated timer - no jitter!
- Up to 4 bi-directional DC motors with individual 8-bit speed selection (so, about 0.5% resolution)
- Up to 2 stepper motors (unipolar or bipolar) with single coil, double coil, interleaved or micro-stepping.
- 4 H-Bridges: L293D chipset provides 0.6A per bridge (1.2A peak) with thermal shutdown protection, 4.5V to 12V
- Pull down resistors keep motors disabled during power-up
- Big terminal block connectors to easily hook up wires (10-22AWG) and power
- Arduino reset button brought up top
- 2-pin terminal block to connect external power, for separate logic/motor supplies
- Tested compatible with Mega, UNO & Duemilanove
- Dimensions: 69mm x 53mm x 14.3mm (2.7in x 2.1in x 0.6in)

Operation :

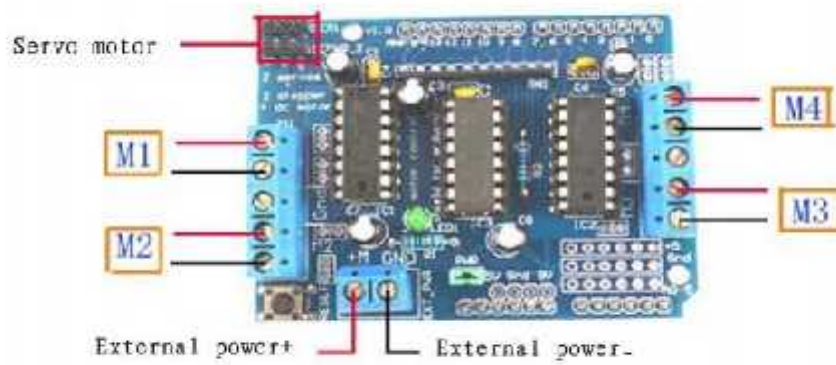
Arduino controller: 1pcs

L293D: 1pcs

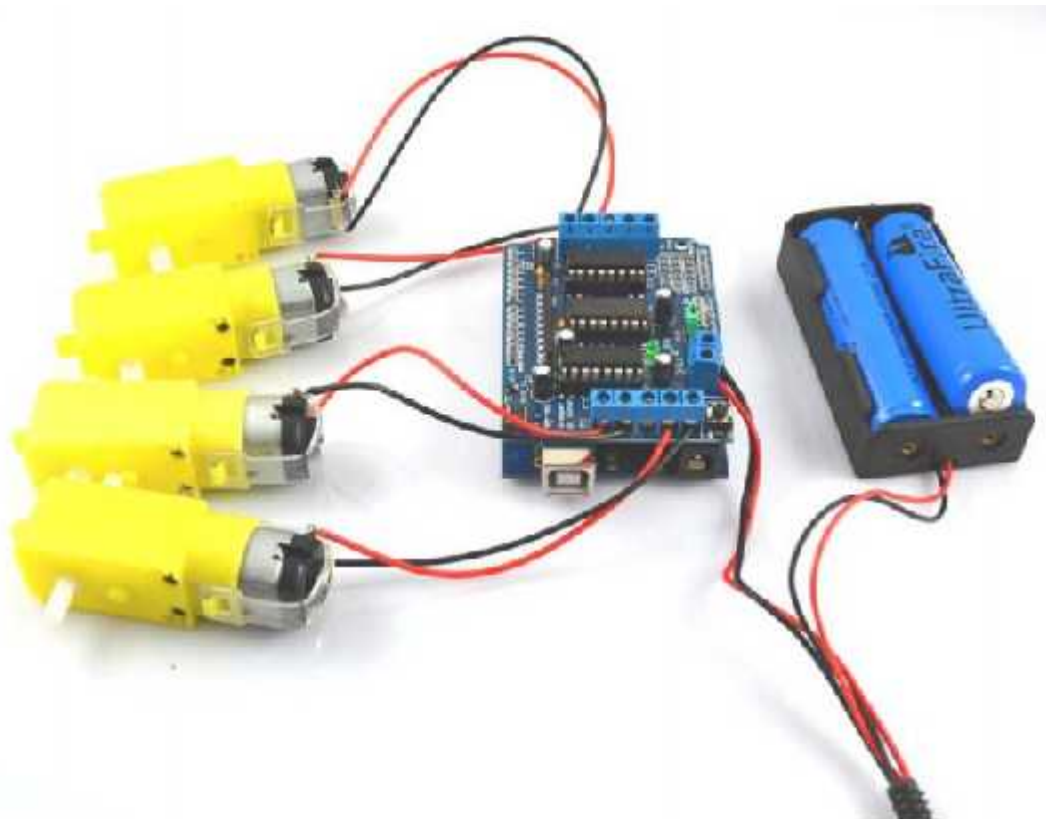
DC motor: 4 pcs

Power supplier 9V: 1pcs

Please connect the devices according to the following drawing:



Drawing



Program source code is as follows :

```
#include <Servo.h>
#define MOTORLATCH 12
#define MOTORCLK 4
#define MOTORENABLE 7
#define MOTORDATA 8
#define MOTOR1_A 2
#define MOTOR1_B 3
#define MOTOR2_A 1
#define MOTOR2_B 4
#define MOTOR3_A 5
#define MOTOR3_B 7
#define MOTOR4_A 0
#define MOTOR4_B 6
#define MOTOR1_PWM 11
#define MOTOR2_PWM 3
#define MOTOR3_PWM 6
#define MOTOR4_PWM 5
#define SERVO1_PWM 10
#define SERVO2_PWM 9
#define FORWARD 1
#define BACKWARD 2
#define BRAKE 3
#define RELEASE 4
Servo servo_1;
Servo servo_2;
void setup()
{
  Serial.begin(9600);
  Serial.println("Simple Adafruit Motor Shield sketch");
  servo_1.attach(SERVO1_PWM);
  servo_2.attach(SERVO2_PWM);
}
void loop()
{
  motor(1, FORWARD, 255);
  motor(2, FORWARD, 255);
  motor(3, FORWARD, 255);
  motor(4, FORWARD, 255);
  delay(2000);
  // Be friendly to the motor: stop it before reverse.
  motor(1, RELEASE, 0);
  motor(2, RELEASE, 0);
```

```

motor(3, RELEASE, 0);
motor(4, RELEASE, 0);
delay(100);
motor(1, BACKWARD, 128);
motor(2, BACKWARD, 128);
motor(3, BACKWARD, 128);
motor(4, BACKWARD, 128);
delay(2000);
motor(1, RELEASE, 0);
motor(2, RELEASE, 0);
motor(3, RELEASE, 0);
motor(4, RELEASE, 0);
delay(100);
}
void motor(int nMotor, int command, int speed)
{
int motorA, motorB;
if (nMotor >= 1 && nMotor <= 4)
{
switch (nMotor)
{
case 1:
motorA = MOTOR1_A;
motorB = MOTOR1_B;
break;
case 2:
motorA = MOTOR2_A;
motorB = MOTOR2_B;
break;
case 3:
motorA = MOTOR3_A;
motorB = MOTOR3_B;
break;
case 4:
motorA = MOTOR4_A;
motorB = MOTOR4_B;
break;
default:
break;
}
switch (command)
{
case FORWARD:
motor_output (motorA, HIGH, speed);

```

```

motor_output (motorB, LOW, -1); // -1: no PWM set
break;
case BACKWARD:
motor_output (motorA, LOW, speed);
motor_output (motorB, HIGH, -1); // -1: no PWM set
break;
case BRAKE:
motor_output (motorA, LOW, 255); // 255: fully on.
motor_output (motorB, LOW, -1); // -1: no PWM set
break;
case RELEASE:
motor_output (motorA, LOW, 0); // 0: output floating.
motor_output (motorB, LOW, -1); // -1: no PWM set
break;
default:
break;
}
}
}
void motor_output (int output, int high_low, int speed)
{
int motorPWM;
switch (output)
{
case MOTOR1_A:
case MOTOR1_B:
motorPWM = MOTOR1_PWM;
break;
case MOTOR2_A:
case MOTOR2_B:
motorPWM = MOTOR2_PWM;
break;
case MOTOR3_A:
case MOTOR3_B:
motorPWM = MOTOR3_PWM;
break;
case MOTOR4_A:
case MOTOR4_B:
motorPWM = MOTOR4_PWM;
break;
default:
speed = -3333;
break;
}
}

```

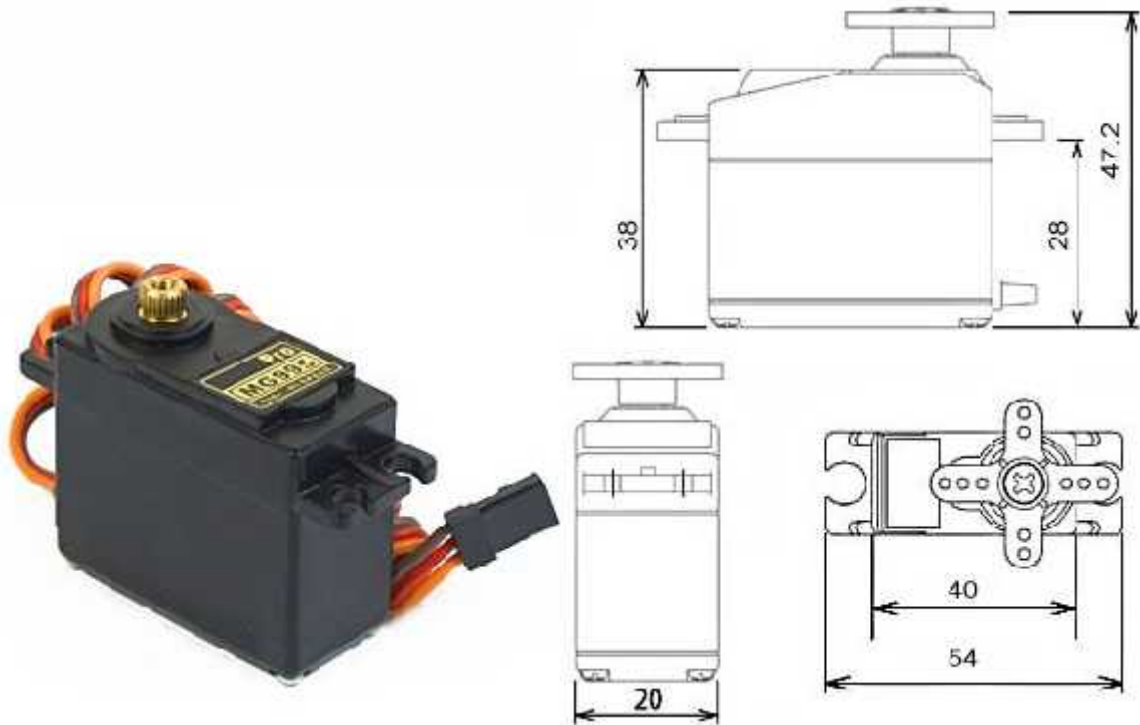


```

if (speed != -3333)
{
shiftWrite(output, high_low);
// set PWM only if it is valid
if (speed >= 0 && speed <= 255)
{
analogWrite(motorPWM, speed);
}
}
}
void shiftWrite(int output, int high_low)
{
static int latch_copy;
static int shift_register_initialized = false;
// Do the initialization on the fly,
// at the first time it is used.
if (!shift_register_initialized)
{
// Set pins for shift register to output pinMode(MOTORLATCH,
OUTPUT); pinMode(MOTORENABLE, OUTPUT);
pinMode(MOTORDATA, OUTPUT); pinMode(MOTORCLK,
OUTPUT);
// Set pins for shift register to default value (low);
digitalWrite(MOTORDATA, LOW); digitalWrite(MOTORLATCH, LOW);
digitalWrite(MOTORCLK, LOW);
// Enable the shift register, set Enable pin Low.
digitalWrite(MOTORENABLE, LOW);
// start with all outputs (of the shift register) low
latch_copy = 0;
shift_register_initialized = true;
}
// The defines HIGH and LOW are 1 and 0.
// So this is valid.
bitWrite(latch_copy, output, high_low);
shiftOut(MOTORDATA, MOTORCLK, MSBFIRST, latch_copy); delayMicroseconds(5); //
For safety, not really needed. digitalWrite(MOTORLATCH, HIGH);
delayMicroseconds(5); // For safety, not really needed.
digitalWrite(MOTORLATCH, LOW);
}

```

MG995 High Speed Metal Gear Dual Ball Bearing Servo



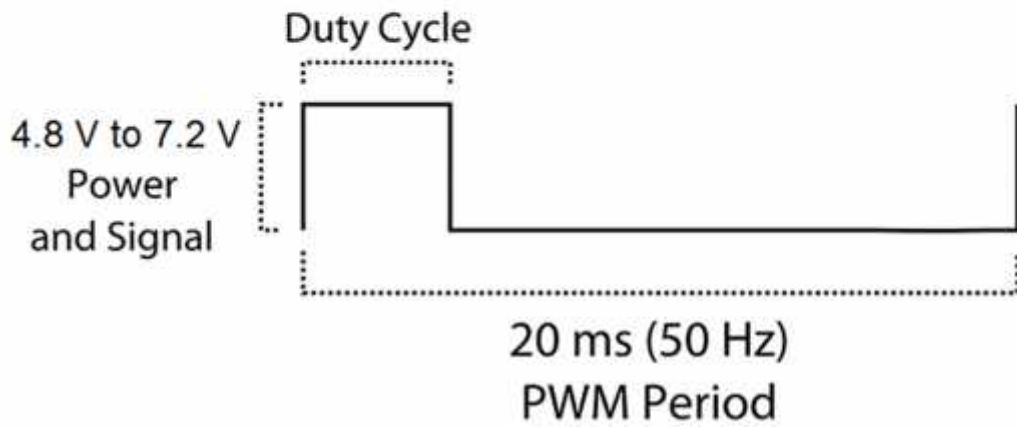
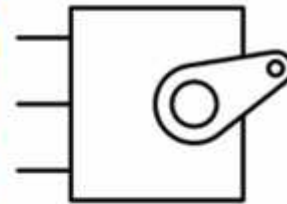
The unit comes complete with 30cm wire and 3 pin 'S' type female header connector that fits most receivers, including Futaba, JR, GWS, Cirrus, Blue Bird, Blue Arrow, Corona, Berg, Spektrum and Hitec.

This high-speed standard servo can rotate approximately 120 degrees (60 in each direction). You can use any servo code, hardware or library to control these servos, so it's great for beginners who want to make stuff move without building a motor controller with feedback & gear box, especially since it will fit in small places. The MG995 Metal Gear Servo also comes with a selection of arms and hardware to get you set up nice and fast!

Specifications

- Weight: 55 g
- Dimension: 40.7 x 19.7 x 42.9 mm approx.
- Stall torque: 8.5 kgf·cm (4.8 V), 10 kgf·cm (6 V)
- Operating speed: 0.2 s/60° (4.8 V), 0.16 s/60° (6 V)
- Operating voltage: 4.8 V a 7.2 V
- Dead band width: 5 μ s
- Stable and shock proof double ball bearing design
- Temperature range: 0 °C – 55 °C

PWM=Orange (⏏)
Vcc = Red (+)
Ground=Brown (-)



Program Arduino ROBOT WIFI

```
//andy_kontrol_robot_wifi
#include <PS2X_lib.h> //PS2X Library

#include <AFMotor.h>
AF_DCMotor motor1(4, MOTOR12_64KHZ); //Jadikan motor 1, 64KHz
AF_DCMotor motor2(3, MOTOR12_64KHZ); //Jadikan motor 2, 64KHz

#include <LiquidCrystal.h>
LiquidCrystal lcd(7, 6, 5, 4, 3, 2);
int analogInput = 0;
float vout = 0.0;
float vin = 0.0;
float vMax = 0.0;
float R1 = 56000.0; // resistance of R1 (100K) -see text!
float R2 = 33000.0; // resistance of R2 (10K) - see text!
int value = 0;

#include <Servo.h>
// membuat nama objek servo untuk pengontrolan servo
Servo myservo;
// variable untuk menyimpan posisi servo
int pos = 0;

//PS2X - Start
PS2X ps2x; // membuat kelas kontrol ps2

int error = 0;
byte type = 0;
byte vibrate = 0;
//PS2X - End

/*
 * menginisialisasi semua. saat berjalan sesekali.
 */
void setup() {
motor1.setSpeed(250); //Set kecepatan motor 250 (Range 0-255)
motor2.setSpeed(250); //Set kecepatan motor 250 (Range 0-255)
Serial.begin(9600); //Kecepatan komunikasi serial

pinMode(analogInput, INPUT); //baterai koding
  lcd.begin(16, 2);
  lcd.print("BATERAI LEVEL");

// objek servo diletakan pada pin 9
myservo.attach(9);

pinMode(led1, OUTPUT); // Tambahan Led1
pinMode(led2, OUTPUT); // Tambahan Led2
//PS2X - Start
Serial.begin(9600);
//Serial.begin(9600); // mengatur Serial library di 9600 bps
error = ps2x.config_gamepad(A1,A3,A4,A2, true, true); //mengatur pin dan settingan: GamePad(clock,
command, attention, data, Pressures?, Rumble?) mendeteksi kesalahan
```

```

if(error == 0)
{
  Serial.println("Kontrol Ditemukan, configured successful");
  Serial.println("Coba semua tombol, X akan menggetarkan kontrol, lebih cepat dengan menekan lebih
keras;");
  Serial.println("Menahan L1 atau R1 akan menampilkan nilai valid stik analog.");
} else if(error == 1)
{
  Serial.println("Kontroller tidak ditemukan, cek kabel/sambungan, lihat readme.txt untuk menghadirkan
penyelesaian masalah.");
} else if(error == 2)
{
  Serial.println("Kontroller ditemukan namun tidak menerima perintah. lihat readme.txt untuk
menghadirkan penyelesaian masalah.");
} else if(error == 3)
{
  Serial.println("Kontrol refusing untuk menerima tekanan pada tombol, bisa jadi tidak mendukung ");
}

//Serial.print(ps2x.Analog(1), HEX);

type = ps2x.readType();
switch(type) {
  case 0:
    Serial.println("Tipe Kontroller Tidak Diketahui");
    break;
  case 1:
    Serial.println("Kontroller DualShock ditemukan");
    break;
  case 2:
    Serial.println("Kontroller GuitarHero ditemukan");
    break;
}

//PS2X - End

}

void loop() {

  // read the value at analog input
  value = analogRead(analogInput);//koding baterai
  vout = (value * 5.0) / 1024.0; // see text
  vin = vout / (R2/(R1+R2));
  if (vin<0.09)
  {
    vin=0.0;//statement to quash undesired reading !
  }
  lcd.setCursor(0, 1);
  lcd.print("TEGANGAN= ");
  lcd.print(vin);
  lcd.print("V");
  delay(500);//koding baterai

  //PS2X - Start
  if(error == 1) //Melewati pengulangan jika kontrol tidak ditemukan
  return;

```

```

if(type == 2) //Kontrol Gitar Hero
    return; //tidak digunakan
else { //Kontroller DualShock
    ps2x.read_gamepad(false, vibrate); //Membaca kontroller dan mengatur besaran motor untuk memutar di kecepatan 'getar'

    //jika analog stik kanan berpindah ke koordinat Y maju-mundur, atau giliran koordinat X kanan-kiri
    if((ps2x.Analog(PSS_RY) < 64) && ((ps2x.Analog(PSS_RX) >= 64) && (ps2x.Analog(PSS_RX) <= 192)))
    {
        //Bergerak Maju
        digitalWrite(led1, HIGH);
        digitalWrite(led2, HIGH);
        Serial.println("FORWARD");
        motor1.run(FORWARD); //Motor kiri maju
        motor2.run(FORWARD); //Motor kanan maju
        digitalWrite(led1, LOW);
    }
    else if((ps2x.Analog(PSS_RY) > 192) && ((ps2x.Analog(PSS_RX) >= 64) && (ps2x.Analog(PSS_RX) <= 192)))
    {
        //Bergerak Mundur
        digitalWrite(led1, LOW);
        digitalWrite(led2, LOW);
        Serial.println("BACKWARD");
        motor1.run(BACKWARD); //Motor kiri mundur
        motor2.run(BACKWARD); //Motor kanan mundur
        digitalWrite(led2, LOW);
    }
    else if((ps2x.Analog(PSS_RX) < 64) && ((ps2x.Analog(PSS_RY) >= 64) && (ps2x.Analog(PSS_RY) <= 192)))
    {
        //Bergerak kekiri
        digitalWrite(led1, HIGH);
        digitalWrite(led2, LOW);
        Serial.println("LEFT");
        motor1.run(BACKWARD); //Motor kiri mundur
        motor2.run(FORWARD); //Motor kanan maju
    }
    else if((ps2x.Analog(PSS_RX) > 192) && ((ps2x.Analog(PSS_RY) >= 64) && (ps2x.Analog(PSS_RY) <= 192)))
    {
        //Bergerak kekanan
        digitalWrite(led1, LOW);
        digitalWrite(led2, HIGH);
        Serial.println("RIGHT");
        motor1.run(FORWARD); //Motor kiri maju
        motor2.run(BACKWARD); //Motor kanan mundur
    }

    //menggunakan tombol up, down, right and left untuk pergerakan
    else if(ps2x.Button(PSB_PAD_UP))//maju
    {
        Serial.print("Up held this hard: ");
        Serial.println(ps2x.Analog(PSAB_PAD_UP), DEC);
        //MENAIKKAN ANTENA 90 DERAJAT
    }
}

```

```

Serial.println("NAIK 90");
for(pos = 0; pos < 90; pos += 1)

// pada posisi 1 derajat
// memberitahu servo untuk pergi ke posisi 'pos'
myservo.write(pos);
// tunggu 15ms untuk pencapaian posisi servo
delay(15);
}

else if(ps2x.Button(PSB_PAD_DOWN))
{
Serial.print("DOWN held this hard: ");
Serial.println(ps2x.Analog(PSAB_PAD_DOWN), DEC);
//Bergerak Turun 90 derajat
Serial.println("TURUN 90");
for(pos = 90; pos >= 1; pos -= 1)

// memberitahu servo untuk pergi ke posisi 'pos'
myservo.write(pos);
// tunggu 15ms untuk pencapaian posisi servo
delay(15);
}

else if(ps2x.Button(PSB_PAD_RIGHT))//kanan
{
Serial.print("Right held this hard: ");
Serial.println(ps2x.Analog(PSAB_PAD_RIGHT), DEC);
//Bergerak kekanan
digitalWrite(led1, LOW);
digitalWrite(led2, HIGH);
Serial.println("RIGHT TURN");
motor1.run(FORWARD); //Motor kiri maju
motor2.run(RELEASE); //Motor kanan BERHENTI
}
else if(ps2x.Button(PSB_PAD_LEFT))
{
Serial.print("LEFT held this hard: ");
Serial.println(ps2x.Analog(PSAB_PAD_LEFT), DEC);
//Bergerak kekiri
digitalWrite(led1, HIGH);
digitalWrite(led2, LOW);
Serial.println("LEFT TURN");
motor1.run(RELEASE); //Motor kiri BERHENTI
motor2.run(FORWARD); //Motor kanan maju
}

else
{
//berhenti
Serial.println("Robot Stopped!");
motor1.run(RELEASE); //Motor kiri BERHENTI
motor2.run(RELEASE); //Motor kanan berhenti
}

}

Serial.print("Stick Values:");

```

```
Serial.print(ps2x.Analog(PSS_LY), DEC); //stik kiri, koordinat Y. pilihan lain: LX, RY, RX
Serial.print(",");
Serial.print(ps2x.Analog(PSS_LX), DEC);
Serial.print(",");
Serial.print(ps2x.Analog(PSS_RY), DEC);
Serial.print(",");
Serial.println(ps2x.Analog(PSS_RX), DEC);
//PS2X - End

}
```




KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
POLITEKNIK NEGERI SRIWIJAYA

Jalan Srijaya Negara, Palembang 30139

Telp. 0711-353414 fax. 0711-355918

Website : www.polisriwijaya.ac.id E-mail : info@polsri.ac.id



SURAT KETERANGAN BEBAS ADMINISTRASI

Yang bertanda tangan di bawah ini menerangkan bahwa,

Nama : Andy Zulkamain
NIM : 061330330960
Kelas : 6TC
Jurusan : Teknik Elektro
Program Studi : Teknik Telekomunikasi

Telah bebas dari segala urusan peminjaman/penggantian alat/buku dan lainnya pada,

No	Kegiatan	Nama	Jabatan	Tanda Tangan	Keterangan
1.	Laboratorium	Sopian ^{Sain} , S.T., M.T	Kepala Laboratorium		
2.	Bengkel	Abdurrahman, S.T., M.Kom	Kepala Bengkel		
3.	Kompensasi	KGS. Herman, A,md	Pengadministrasi Jurusan		
4.	Perpustakaan Polsri	Hj. Nelli Lingga Yunara, S.pd	Kepala Perpustakaan		
5.	Perpustakaan Jurusan	Tusk Yarni, S.H	Kasi Perpustakaan		
6.	Pembimbingan Laporan Akhir	Ir. Ibnu Ziad, M.T	Pembimbing I		
		Nasron, S.T., M.T	Pembimbing II		

Demikian surat keterangan ini agar dapat digunakan seperlunya.

Palembang, Agustus 2016
Ketua Jurusan Teknik Elektro

(Yudi Wijanarko, S.T., M.T)
NIP. 196705111992031003



KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
POLITEKNIK NEGERI SRIWIJAYA

Jalan Srijaya Negara, Palembang 30139
Telp. 0711-353414 fax. 0711-355918

Website : www.polisriwijaya.ac.id E-mail : info@polsri.ac.id



BUKTI PENYERAHAN HASIL KARYA/RANCANG BANGUN

Pada hari ini tanggal bulan tahun telah diserahkan seperangkat karya/rancang bangun kepada Jurusan Program Studi di Politeknik Negeri Sriwijaya,

Nama Perangkat	Spesifikasi
ROBOT WIFI MENGGUNAKAN ANTENA WAJANBOLIC DENGAN KONTROL JOYSTIK UNTUK Mencari SINYAL WIFI TERKUAT BERBASIS ATMEGA 328	Antena Wajanbolic, Arduino Duemilanove, USB Adapter TL-WN727N, Router Tenda N301, Driver Motor Adafruit L293D, Baterai Litium-Ion 20V 1500Mah


Hasil karya/rancang bangun dari,

Nama	NIM	Nama Pembimbing
ANDY ZULKARNAIN	061330330960	Ir. Ibnu Ziad, M.T Nasron, S.T., M.T

Yang menerima *),

Sopian Soim, S.T., M.T
NIP. 197103142001121001

Palembang, Agustus 2016
Yang menyerahkan **),


Andy Zulkarnain
NIM. 061330330960

Mengetahui,
Ketua Jurusan Teknik Elektro

Yudi Wijanarko, S.T., M.T
NIP. 196705111992031003

*) pejabat Jurusan/PS yang ditunjuk (Kepala Lab./Bengkel atau Kepala Seksi)
**) perwakilan mahasiswa dari pembuat karya/rancang bangun.



KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
POLITEKNIK NEGERI SRIWIJAYA
 Jalan Sriwijaya Negara, Palembang 30139
 Telp. 0711-353414 Fax. 0711-355918
 Website : www.polisriwijaya.ac.id E-mail : info@polsri.ac.id



PELAKSANAAN REVISI LAPORAN AKHIR

Mahasiswa berikut,

Nama : Andy Zulkarnain
 NIM : 061330330960
 Jurusan/Program Studi : Teknik Elektro/ Teknik Telekomunikasi
 Judul Laporan Akhir : Robot Wifi Menggunakan Antena Wajanbolic Dengan Kontrol Joystik Untuk Mencari Sinyal Wifi Terkuat Berbasis ATMega 328

Telah melaksanakan revisi terhadap Laporan Akhir yang diujikan pada hari Kamis tanggal 4 bulan Agustus tahun 2016 Pelaksanaan revisi terhadap Laporan Akhir tersebut telah disetujui oleh Dosen Penguji yang memberikan revisi:

No.	Komentar	Nama Dosen Penguji ^{*)}	Tanggal	Tanda Tangan
1.	Acc	Nasron S.T.,M.T NIP. 196808221993031001	24/8-2016	
2.	Acc	Ir.H. Ibnu Ziad M.T NIP. 196005161990031001	10/8-2016	
3.	Acc	Ir. Jon Endri, M.T NIP. 196201151993031001	24/8-2016	
4.	Acc	R.A. Halimatassa'diyah,S.T.,M.Kom NIP. 197406022005012002	19/8-2016	
5.	Acc	Suzanzefi, S.T.,M.Kom NIP. 197709252005012003	23/8-2016	
6.	Acc	Eka Susanti, S.T.,M.Kom NIP. 197812172000122001	31/8/16	

Palembang, 31 Agustus 2016

Ketua Penguji ^{**)},

(Nasron S.T.,M.T)
 NIP. 196808221993031001

Catatan:

*) Dosen penguji yang memberikan revisi saat ujian laporan akhir.

**) Dosen penguji yang ditugaskan sebagai Ketua Penguji saat ujian LA.
 Lembaran pelaksanaan revisi ini harus dilampirkan dalam Laporan Akhir.



KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
POLITEKNIK NEGERI SRIWIJAYA

Jalan Srijaya Negara, Palembang 30139

Telp. 0711-353414 Fax. 0711-355918

Website : www.polisriwijaya.ac.id E-mail : info@polsri.ac.id



KESEPAKATAN BIMBINGAN LAPORAN AKHIR (LA)

Kami yang bertanda tangan di bawah ini,

Pihak Pertama :

Nama : Andy Zulkarnain
NIM : 061330330960
Jurusan : Teknik Elektro
Program Studi : Teknik Telekomunikasi D3

Pihak Kedua :

Nama : Ir. H. Ibnu Ziad, M.T.
NIP : 196005161990031001
Jurusan : Teknik Elektro
Program Studi : Teknik Telekomunikasi

Pada hari ini Kamis tanggal 21-7-2016 telah sepakat untuk melakukan konsultasi bimbingan Laporan Akhir.

Konsultasi bimbingan sekurang-kurangnya 1 (satu) kali dalam satu minggu. Pelaksanaan bimbingan pada setiap hari Kamis pukul 13:00, tempat di Politeknik Negeri Sriwijaya.

Demikianlah kesepakatan ini dibuat dengan penuh kesadaran guna kelancaran penyelesaian Laporan Akhir.

Pihak Pertama,

(Andy Zulkarnain)
NIM 061330330960

Palembang, 21-7-2016

Pihak Kedua,

(Ir. H. Ibnu Ziad, M.T)
NIP 196005161990031001

Mengetahui,
Ketua Jurusan Teknik Elektro

(Yudi Wijanarko, ST.M.T.)
NIP 196705111992031003



KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
POLITEKNIK NEGERI SRIWIJAYA

Jalan Srijaya Negara, Palembang 30139

Telp. 0711-353414 Fax. 0711-355918

Website : www.polisriwijaya.ac.id E-mail : info@polsri.ac.id



KESEPAKATAN BIMBINGAN LAPORAN AKHIR (LA)

Kami yang bertanda tangan di bawah ini,

Pihak Pertama :

Nama : Andy Zulkarnain
NIM : 061330330960
Jurusan : Teknik Elektro
Program Studi : Teknik Telekomunikasi D3

Pihak Kedua :

Nama : Nasron, S.T., M.T.
NIP : 196808221993031001
Jurusan : Teknik Elektro
Program Studi : Teknik Telekomunikasi

Pada hari ini Juni tanggal 20 Juni 2016 telah sepakat untuk melakukan konsultasi bimbingan Laporan Akhir.

Konsultasi bimbingan sekurang-kurangnya 1 (satu) kali dalam satu minggu. Pelaksanaan bimbingan pada setiap hari Juni pukul 10.00, tempat di Politeknik Negeri Sriwijaya.

Demikianlah kesepakatan ini dibuat dengan penuh kesadaran guna kelancaran penyelesaian Laporan Akhir.

Pihak Pertama,

(Andy Zulkarnain)
NIM 061330330960

Palembang, 20 Juni 2016

Pihak Kedua,

(Nasron, S.T., M.T.)
NIP 196808221993031001

Mengetahui,
Ketua Jurusan Teknik Elektro

(Yudi Wijanarko, ST.M.T.)
NIP 196705111992031003








KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
POLITEKNIK NEGERI SRIWIJAYA
 Jalan Srijaya Negara, Palembang 30139
 Telp. 0711-353414 Fax. 0711-355918
 Website : www.polsriwijaya.ac.id E-mail : info@polsri.ac.id


LEMBAR BIMBINGAN LAPORAN AKHIR


Lembar : 1

Nama : Andy Zulkamain
 NIM : 081330330960
 Jurusan/Program Studi : Teknik Elektro/Teknik Telekomunikasi
 Judul Laporan Akhir : Robot Wifi Menggunakan Antena Wajanbofic Dengan Kontrol Joystik Untuk Mencari Sinyal Wifi Terkuat Berbasis ATmega328
 Pembimbing I : Ir. H. Ibnu Ziad, M.T

No.	Tanggal	Uraian Bimbingan	Tanda Tangan Pembimbing
1.	20/6-2016	Konsultasi Laporan, Garis Judul	
2.	23/6-2016	Konsultasi Alat dan laporan	
3.	27/6-2016	Konsultasi Alat, pengurusan	
4.	21/7-2016	Lengkapi Caporan	
5.	25/7-2016	Pertahankan Caporan sudah lengkap	
6.	27/7-2016	Caporan sudah diperbaiki tapi harus diperiksa lagi	
7.	26/7-2016	Revisi BAB III & IV	

No.	Tanggal	Uraian Bimbingan	Tanda Tangan Pembimbing
8.	26/7-2011	Acc bab <u>III</u> & <u>IV</u>	
9.	26/7-2011	Revisi bab <u>V</u>	
10.	26/7-2011	Acc bab <u>V</u>	
11.	26/7-2016	Revisi lampiran antara lain kertas gambar, dan lain-lain	
12.	27/7-2016	Si'ap IKUT ujian LA	

Palembang, 2016
Kepala Program Studi Teknik Telekomunikasi


(Ciksadan, S.T., M.Kom)
NIP. 196809071993031003

Catatan:

*) melingkari angka yang sesuai.

Ketua Jurusan/Ketua Program Studi harus memeriksa jumlah pelaksanaan bimbingan sesuai yang dipersyaratkan dalam Pedoman Laporan Akhir sebelum menandatangani lembar bimbingan ini.

Lembar pembimbingan LA ini harus dilampirkan dalam Laporan Akhir.



KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
POLITEKNIK NEGERI SRIWIJAYA

Jalan Srijaya Negara, Palembang 30139

Telp. 0711-353414 Fax. 0711-355918

Website : www.polsriwijaya.ac.id E-mail : info@polsri.ac.id








LEMBAR BIMBINGAN LAPORAN AKHIR

Lembar : 1

Nama : Andy Zulkarnain
 NIM : 061330330960
 Jurusan/Program Studi : Teknik Elektro/Teknik Telekomunikasi
 Judul Laporan Akhir : Robot Wifi Menggunakan Antena Wajanbolic Dengan Kontrol Joystik Untuk Mencari Sinyal Wifi Terkuat Berbasis ATmega328
 Pembimbing II : Nasron, S.T.,M.T.

No.	Tanggal	Uraian Bimbingan	Tanda Tangan Pembimbing
1.	17/6-2016	konsultasi Alat	
2.	20/6-2016	konsultasi Alat, Ganti mikrokontroler	
3.	23/6-2016	konsultasi Laporan, Tambah Judul	
4.	24/6-2016	Bab 1 Revisi	
5.	27/6-2016	Bab 1 Acc	
6.	28/6-2016	Bab 2 Revisi	
7.	29/6-2016	Bab 2 Acc	

No.	Tanggal	Uraian Bimbingan	Tanda Tangan Pembimbing
8.	18/7-2016	Bab 3 Revisi	
9.	19/7-2016	Bab 3 Acc	
10.	20/7-2016	Bab 4 & 5 Revisi	
11.	21/7-2016	Bab 4 & 5 Acc	
12.	22/7-2016	Acc- siap untuk ujian LA	

Palembang,
Kepala Program Studi Teknik Telekomunikasi


(Ciksadin, S.T., M.Kom)
NIP. 196809071993031003

Catatan:

*) melingkari angka yang sesuai.

Ketua Jurusan/Ketua Program Studi harus memeriksa jumlah pelaksanaan bimbingan sesuai yang dipersyaratkan dalam Pedoman Laporan Akhir sebelum menandatangani lembar bimbingan ini.

Lembar pembimbingan LA ini harus ditampirkan dalam Laporan Akhir.

Palembang, Juli 2016

Kepada
Yth. Kepala Laboratorium
Teknik Telekomunikasi
di
Tempat

Saya yang bertanda tangan dibawah ini :

Nama : Andy Zulkarnain

NIM : 0613 3033 0960

Kelas : 6 TC

Judul : Robot Wifi Menggunakan Antena Wajanbolic Dengan Kontrol
Joystik Untuk Mencari Sinyal Wifi Terkuat Berbasis ATmega328



Dengan ini mengajukan permohonan untuk menggunakan laboratorium serta meminjam beberapa peralatan praktikum yang tersedia dilaboratorium.

Peralatan tersebut digunakan sebagai alat untuk mengambil data serta syarat untuk menyelesaikan laporan akhir jurusan Teknik Elektro Program Studi Teknik Telekomunikasi. Untuk kepentingan pengambilan data mohon kesediaan Bapak/ Ibu Pembimbing untuk mendampingi.

Yang Bersangkutan



Andy Zulkarnain
NPM. 061330330960

No.	Tanggal	Pembimbing I	Pembimbing II	Keterangan
1.	20-7-2016			Pengukuran
2.				
3.				
4.				
5.				
6.				
7.				
8.				
9.				
10.				
11.				
12.				
13.				

Mengetahui,

Ketua Program Studi

Kepala Lab. Telekomunikasi



Ciksadan, S.T., M.Kom
NIP. 196809071993031003

Sopian Soim, S.T., M.T
NIP.197103142001121001

No. Dok : F-PBM-18

Tgl. Berlaku : 13 Desember 2010

No. Rev : 00



KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
POLITEKNIK NEGERI SRIWIJAYA
Jalan Srijaya Negara, Palembang 30139
Telp. 0711-353414 fax. 0711-355918
Website : www.polisriwijaya.ac.id E-mail : info@polsri.ac.id



REKOMENDASI UJIAN LAPORAN AKHIR (LA)

Pembimbing Laporan Akhir memberikan rekomendasi kepada,

Nama : Andy Zulkarnain
NIM : 061330330960
Jurusan/Program Studi : Teknik Elektro/Teknik Telekomunikasi D3
Judul Laporan Akhir : Robot Wifi Menggunakan Antena
Wajanbolic Dengan Kontrol Joystik Untuk
Mencari Sinyal Wifi Terkuat Berbasis
ATMega 328

Mahasiswa tersebut telah memenuhi persyaratan dan dapat mengikuti Ujian
Laporan Akhir (LA) pada Tahun Akademik 2015/2016

Palembang, 27 Juli 2016

Pembimbing I,

(Ir. H. Ibnu Ziad, M.T)
NIP 196005161990031001

Pembimbing II,

(Nasron, S.T., M.T)
NIP 196808221993031001