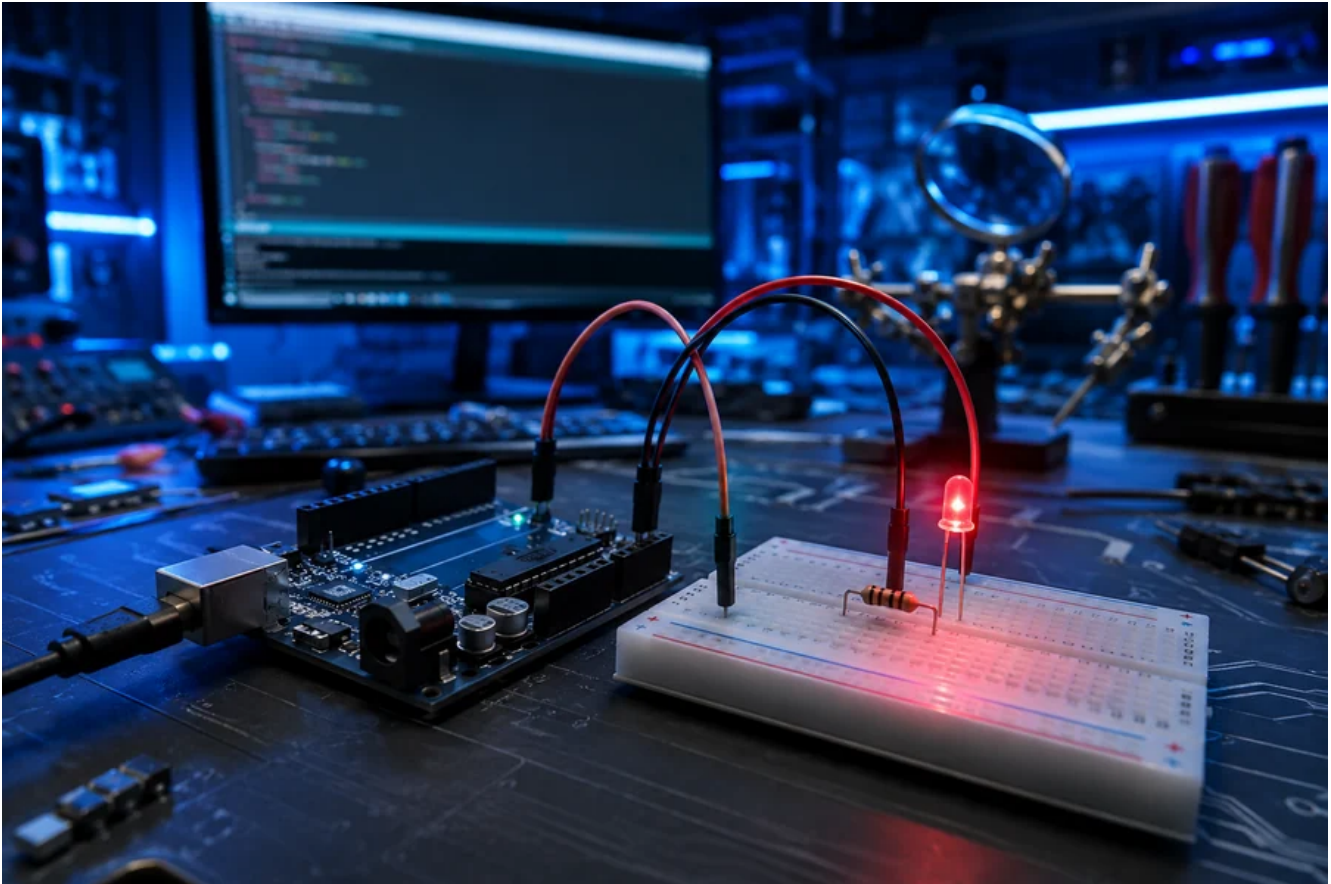


WolfieWeb Complete Arduino Lab Manual

Printable pinout, wiring, code, build steps, expected results, and troubleshooting for every Arduino tutorial on the WolfieWeb Arduino page.

Arduino LED Blink Lab Manual



Goal: Learn digital output, LED polarity, current limiting, and timing.

Pin Codes and Wiring Map

Part / Lead	Arduino Pin / Connection	Purpose
Arduino 5V	Breadboard + rail	Optional rail power
Arduino GND	Breadboard - rail	Common ground
D13 or D8	220 ohm resistor -> LED anode	Output control
LED cathode	GND rail	Return path

Step-by-Step Build Instructions

Step 1: Place the LED across two separate breadboard rows. The longer leg is the anode/positive side.

Step 2: Put the 220 ohm resistor in series with the LED. Do not skip it; the resistor limits current.

Step 3: Connect D13 to the resistor side feeding the LED anode. Connect the LED cathode to GND.

Step 4: Select the correct board and COM port in Arduino IDE. Upload the blink sketch.

Step 5: Change delay(500) to delay(100) and delay(1000) to prove the code controls timing.

Expected Result

The LED turns on and off repeatedly. If it stays dark, flip the LED, check the resistor path, and confirm the selected COM port.

Troubleshooting

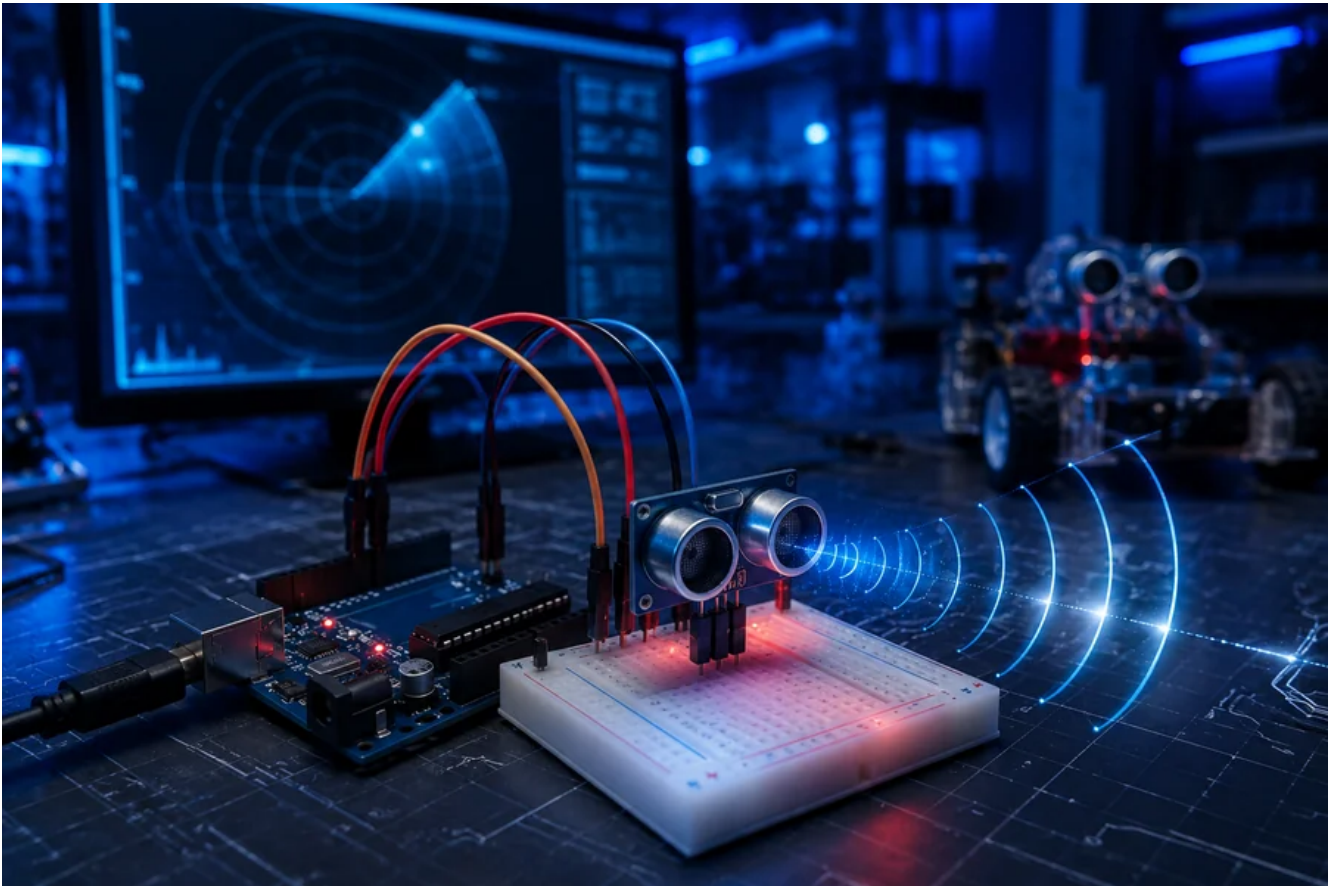
Problem	What to check
Upload fails	Board/port is wrong, USB cable is charge-only, or another serial monitor is open.
LED always on	Output pin may be connected directly to 5V instead of a controlled pin.
LED never lights	LED reversed, resistor not in series, or no ground return.

Arduino Code

```
void setup() {  
  pinMode(13, OUTPUT);  
}  
  
void loop() {  
  digitalWrite(13, HIGH);  
  delay(500);  
  digitalWrite(13, LOW);  
  delay(500);  
}
```

Scan the QR code on the package README or visit www.wolfieweb.com/works01.html for the live tutorial page.

Arduino HC-SR04 Ultrasonic Sensor Lab Manual



Goal: Measure distance using trigger and echo timing. This is the foundation of obstacle detection.

Pin Codes and Wiring Map

Part / Lead	Arduino Pin / Connection	Purpose
VCC	5V	Sensor power
GND	GND	Common ground
TRIG	D9	Arduino sends 10 microsecond pulse
ECHO	D10	Arduino measures pulse return time

Step-by-Step Build Instructions

Step 1: Mount the HC-SR04 facing forward with the two round transducers unobstructed.

Step 2: Wire VCC to 5V and GND to GND first.

Step 3: Wire TRIG to D9 and ECHO to D10. Keep the sensor wires short for stable readings.

Step 4: Upload the distance sketch and open Serial Monitor at 9600 baud.

Step 5: Move your hand from 10 cm to 50 cm away and verify readings change smoothly.

Expected Result

The serial monitor prints distance values in centimeters. Values should shrink as your hand moves closer.

Troubleshooting

Problem	What to check
Always reads 0	TRIG/ECHO are swapped, ECHO pin is wrong, or sensor is not powered.
Random spikes	Sensor is angled, target is soft fabric, or wires are loose.
No serial output	Serial Monitor baud rate is not set to 9600.

Arduino Code

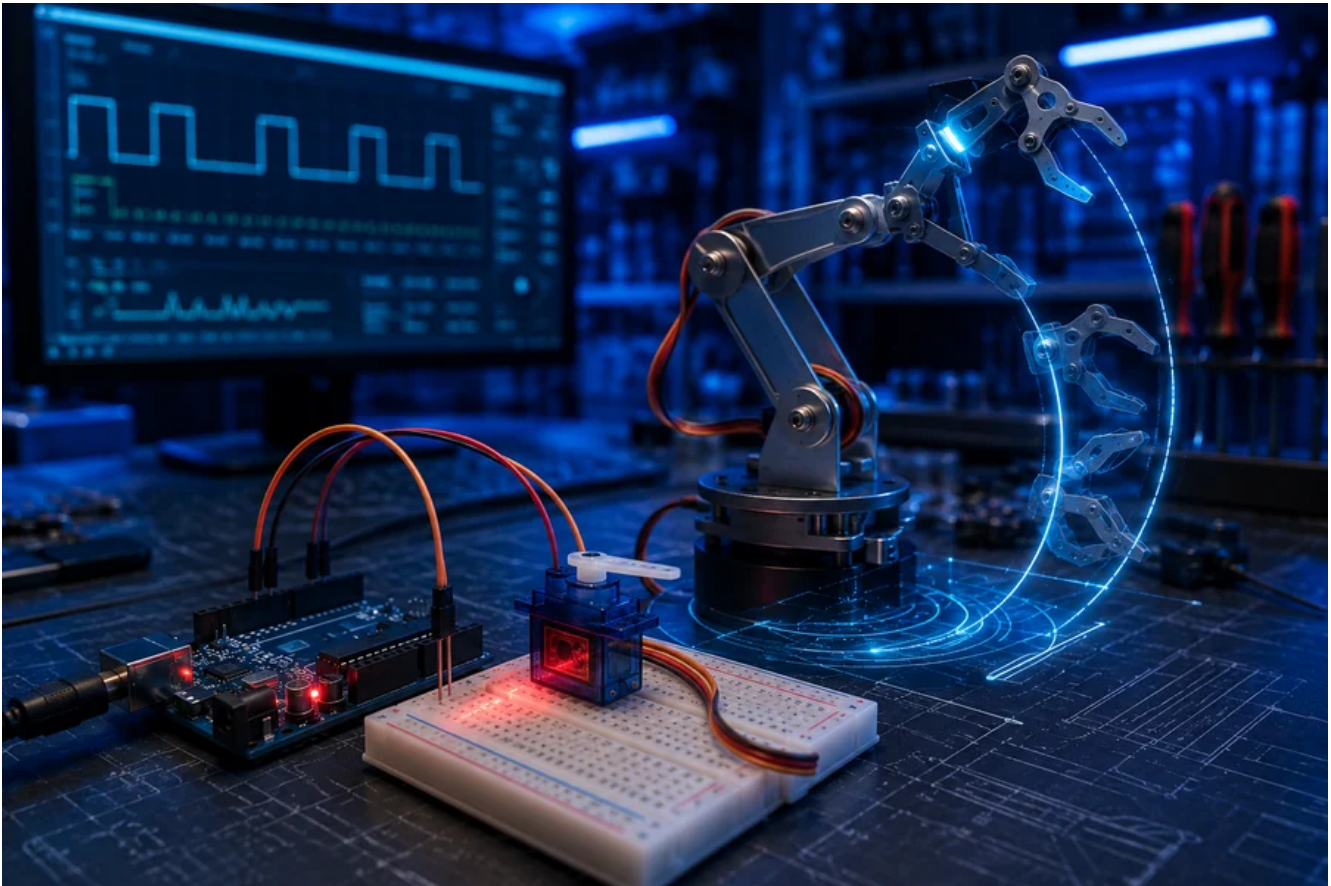
```
const int trigPin = 9;
const int echoPin = 10;

void setup() {
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  Serial.begin(9600);
}

void loop() {
  digitalWrite(trigPin, LOW); delayMicroseconds(2);
  digitalWrite(trigPin, HIGH); delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  long duration = pulseIn(echoPin, HIGH);
  float distanceCm = duration * 0.0343 / 2;
  Serial.println(distanceCm);
  delay(250);
}
```

Scan the QR code on the package README or visit www.wolfieweb.com/works01.html for the live tutorial page.

Arduino SG90 Servo Motor Lab Manual



Goal: Control motion by commanding servo angles from code.

Pin Codes and Wiring Map

Part / Lead	Arduino Pin / Connection	Purpose
Servo red	5V	Power
Servo brown/black	GND	Ground
Servo orange/yellow	D9	Signal/control
External 5V supply GND	Arduino GND	Required if using separate servo power

Step-by-Step Build Instructions

Step 1: Connect servo ground before signal. A missing ground causes jitter and random motion.

Step 2: Connect servo signal to D9. Use the Servo library instead of trying to manually time pulses.

Step 3: Upload a sweep sketch and keep fingers clear of the horn.

Step 4: Test 0, 90, and 180 degrees. If the mechanism binds, lower the angle range.

Step 5: If the Arduino resets, power the servo from a separate 5V supply and tie grounds together.

Expected Result

The servo moves smoothly between commanded angles. A light hum is normal; violent buzzing means binding or weak power.

Troubleshooting

Problem	What to check
Servo twitches	Power is weak, signal wire loose, or ground not shared.
Servo does not move	Signal pin does not match code or servo connector is reversed.
Arduino resets	Servo current draw is too high for USB power.

Arduino Code

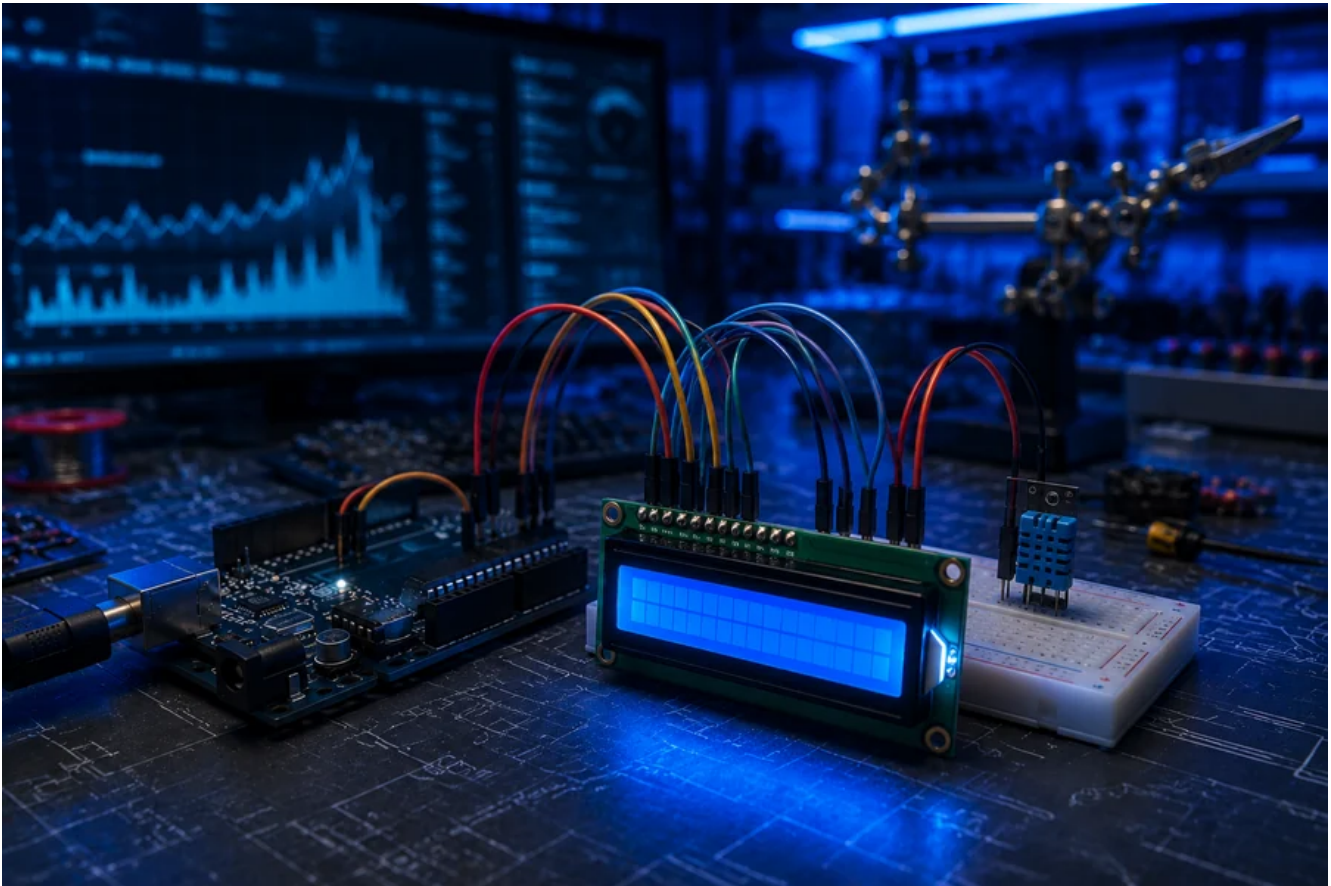
```
#include <Servo.h>
Servo myServo;

void setup() {
  myServo.attach(9);
}

void loop() {
  myServo.write(0); delay(700);
  myServo.write(90); delay(700);
  myServo.write(180); delay(700);
}
```

Scan the QR code on the package README or visit www.wolfieweb.com/works01.html for the live tutorial page.

Arduino LCD Display Lab Manual



Goal: Display live values instead of hiding everything in the serial monitor.

Pin Codes and Wiring Map

Part / Lead	Arduino Pin / Connection	Purpose
LCD VSS	GND	Ground
LCD VDD	5V	Power
LCD VO	10K pot center pin	Contrast
LCD RS	D12	Register select
LCD E	D11	Enable
LCD D4	D5	Data
LCD D5	D4	Data
LCD D6	D3	Data
LCD D7	D2	Data

Step-by-Step Build Instructions

Step 1: Wire power and contrast first. If contrast is wrong, a working LCD can look dead.

Step 2: Wire RS, E, D4, D5, D6, and D7 exactly as the code defines.

Step 3: Install or include the LiquidCrystal library.

Step 4: Upload the test sketch and look for the first line text.

Step 5: Use `lcd.setCursor(0,1)` to write sensor data on the second line.

Expected Result

The LCD shows the title on line one and live values on line two. Adjust the contrast if the screen is blank or filled with blocks.

Troubleshooting

Problem	What to check
Blank screen	Contrast pin not adjusted, power missing, or backlight not connected.
Gibberish text	Data pins do not match the code.
Only blocks show	LCD initialized wrong or contrast set too high.

Arduino Code

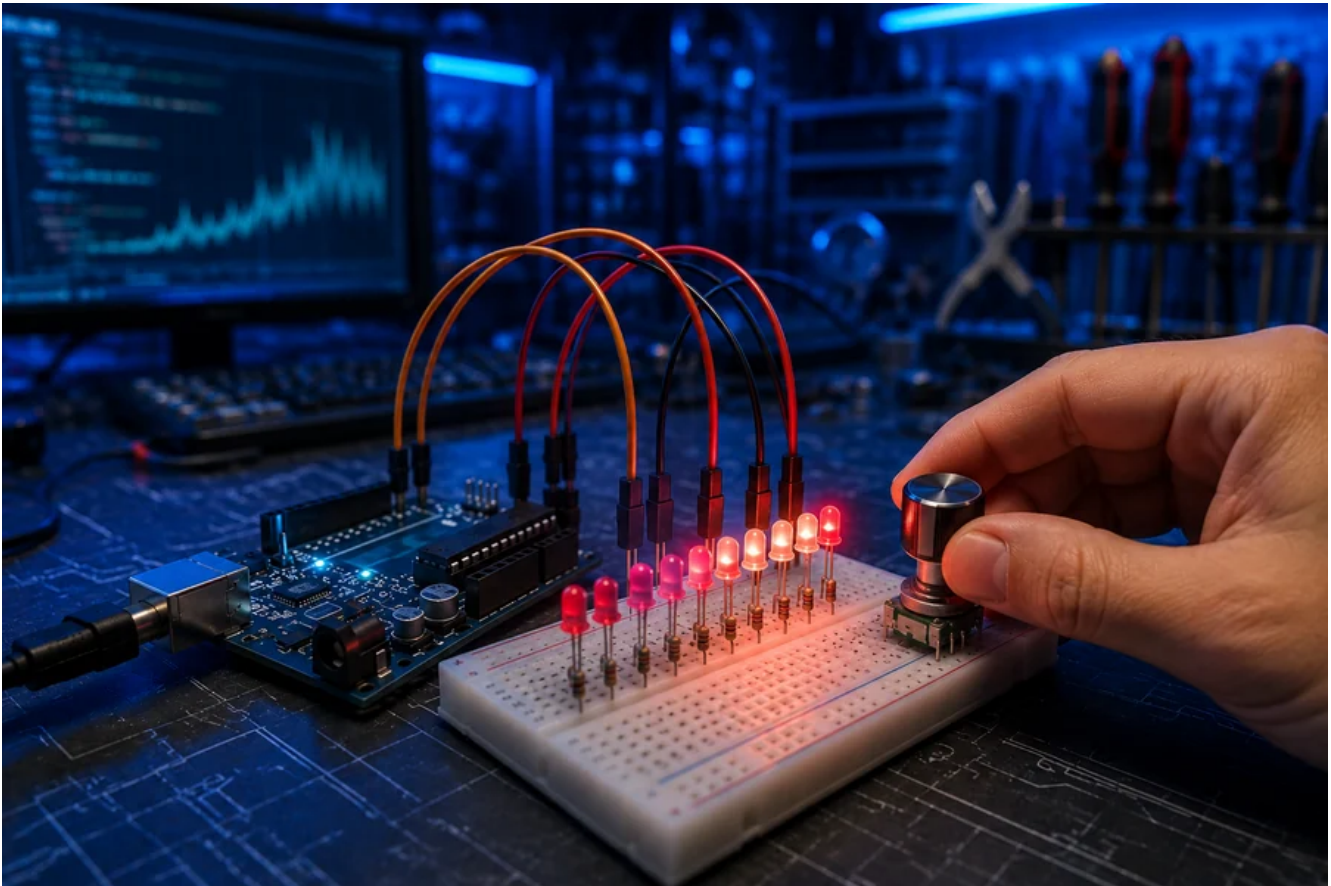
```
#include <LiquidCrystal.h>
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

void setup() {
  lcd.begin(16, 2);
  lcd.print("Temp Monitor");
}

void loop() {
  int sensorValue = analogRead(A0);
  lcd.setCursor(0, 1);
  lcd.print("Value: ");
  lcd.print(sensorValue);
  lcd.print(" ");
  delay(500);
}
```

Scan the QR code on the package README or visit www.wolfieweb.com/works01.html for the live tutorial page.

Arduino Potentiometer Analog Input Lab Manual



Goal: Read a physical knob and turn it into usable values from 0 to 1023.

Pin Codes and Wiring Map

Part / Lead	Arduino Pin / Connection	Purpose
Pot side pin 1	5V	Reference voltage
Pot side pin 2	GND	Ground reference
Pot center pin	A0	Analog signal
Optional LED PWM pin	D9	Use mapped value for brightness later

Step-by-Step Build Instructions

Step 1: Place the potentiometer so each leg lands on a separate breadboard row.

Step 2: Connect one outer leg to 5V and the other outer leg to GND.

Step 3: Connect the center/wiper leg to A0.

Step 4: Upload the analogRead sketch and open Serial Monitor at 9600 baud.

Step 5: Turn the knob slowly. Use map() later to convert 0-1023 into brightness, speed, or angle.

Expected Result

Serial Monitor values move smoothly from near 0 to near 1023 as the knob turns.

Troubleshooting

Problem	What to check
Value stuck at 0	Center pin may be wired to GND or A0 is disconnected.
Value stuck at 1023	Center pin may be wired to 5V.
Values jump wildly	Loose breadboard connection or no shared ground.

Arduino Code

```
void setup() {  
  Serial.begin(9600);  
}  
  
void loop() {  
  int knobValue = analogRead(A0);  
  Serial.println(knobValue);  
  delay(200);  
}
```

Scan the QR code on the package README or visit www.wolfieweb.com/works01.html for the live tutorial page.

Arduino RFID Smart Lock Lab Manual



Goal: Build a simple access-control system using card ID logic, LED feedback, a buzzer, and servo motion.

Pin Codes and Wiring Map

Part / Lead	Arduino Pin / Connection	Purpose
MFRC522 VCC	3.3V	Do not use 5V unless your board is 5V tolerant
MFRC522 GND	GND	Ground
MFRC522 RST	D9	Reset
MFRC522 SDA/SS	D10	SPI select
MFRC522 MOSI	D11	SPI data
MFRC522 MISO	D12	SPI data
MFRC522 SCK	D13	SPI clock
Servo signal	D6	Lock control
Green LED	D4 via 220 ohm	Approved
Red LED	D5 via 220 ohm	Denied
Buzzer +	D3	Audio feedback

Step-by-Step Build Instructions

Step 1: Wire only the RFID module first and run a UID reader test.

Step 2: Scan the approved card and copy the UID exactly from Serial Monitor.

Step 3: Add the servo and test locked/unlocked positions before adding access logic.

Step 4: Add green LED, red LED, and buzzer feedback.

Step 5: Load the full lock sketch. Approved UID should unlock; unknown cards should be denied.

Step 6: Mount the servo beside a small latch. Keep the first mechanical version simple and reliable.

Expected Result

Approved card lights green, chirps the buzzer, rotates the servo open, waits, then relocks. Wrong card lights red and stays locked.

Troubleshooting

Problem	What to check
No card detected	Check 3.3V, SDA pin 10, and SPI pins 11/12/13.
Correct card denied	UID string does not match exactly. Copy spacing and capitalization.
RFID fails when servo moves	Servo noise or current draw is disturbing power. Use separate servo power with shared ground.

Arduino Code

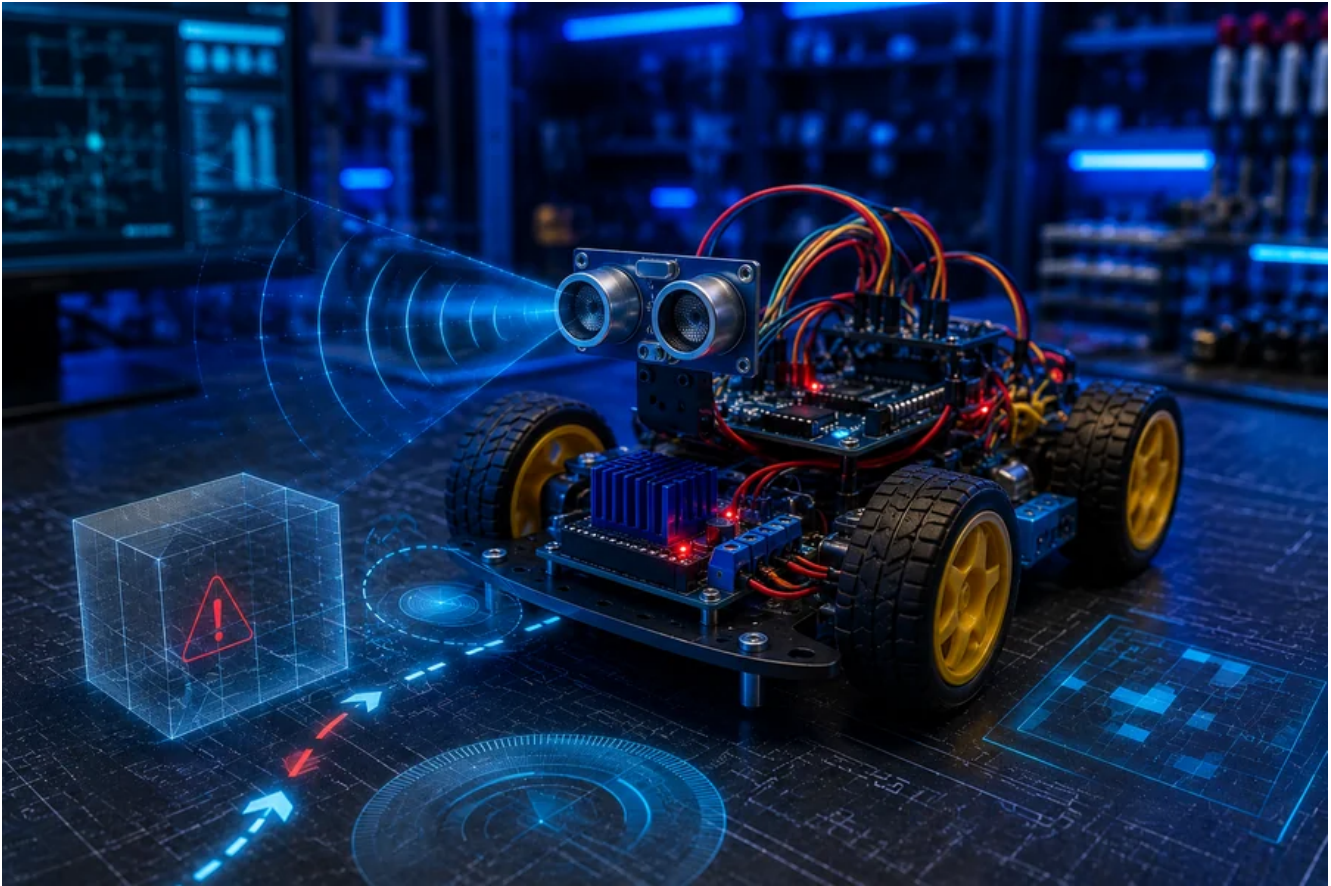
```
#include <SPI.h>
#include <MFRC522.h>
#include <Servo.h>
#define SS_PIN 10
#define RST_PIN 9
MFRC522 rfid(SS_PIN, RST_PIN);
Servo lockServo;
String approvedUID = "DE AD BE EF";

void setup() {
  Serial.begin(9600); SPI.begin(); rfid.PCD_Init();
  lockServo.attach(6); lockServo.write(0);
  pinMode(3, OUTPUT); pinMode(4, OUTPUT); pinMode(5, OUTPUT);
}

void loop() {
  if (!rfid.PICC_IsNewCardPresent()) return;
  if (!rfid.PICC_ReadCardSerial()) return;
  String cardUID = "";
  for (byte i=0; i<rfid.uid.size; i++) {
    if (rfid.uid.uidByte[i] < 0x10) cardUID += "0";
    cardUID += String(rfid.uid.uidByte[i], HEX);
    if (i < rfid.uid.size-1) cardUID += " ";
  }
  cardUID.toUpperCase();
  if (cardUID == approvedUID) { digitalWrite(4,HIGH); tone(3,1200,150); lockServo.write(90); delay(3000);
  lockServo.write(0); digitalWrite(4,LOW); }
  else { digitalWrite(5,HIGH); tone(3,300,400); delay(700); digitalWrite(5,LOW); }
  rfid.PICC_HaltA(); rfid.PCD_StopCrypto1();
}
```

Scan the QR code on the package README or visit www.wolfieweb.com/works01.html for the live tutorial page.

Arduino Obstacle Avoiding Robot Car Lab Manual



Goal: Combine sensor input, motor control, power management, and decision logic into a moving robot.

Pin Codes and Wiring Map

Part / Lead	Arduino Pin / Connection	Purpose
HC-SR04 VCC	5V	Sensor power
HC-SR04 GND	GND	Ground
HC-SR04 TRIG	D9	Trigger
HC-SR04 ECHO	D10	Echo
L298N ENA	D5	Left speed PWM
L298N IN1	D7	Left direction
L298N IN2	D8	Left direction
L298N IN3	D11	Right direction
L298N IN4	D12	Right direction
L298N ENB	D6	Right speed PWM
Motor battery +	L298N VIN/12V	Motor power
Battery -	L298N GND + Arduino GND	Shared ground

Step-by-Step Build Instructions

Step 1: Assemble the chassis and mount motors firmly. Wheels should spin freely.

Step 2: Mount the L298N close to the motors. Keep motor wires short and secure.

Step 3: Wire the motors to OUT1/OUT2 and OUT3/OUT4. Lift the car and run a motor test.

Step 4: If one wheel spins backward, swap that motor wires or reverse direction in code.

Step 5: Mount the ultrasonic sensor at the front like robot eyes, level with the path.

Step 6: Upload a distance-only test before the full robot code. Do not debug motors and sensors at the same time.

Step 7: Upload the obstacle code. Start with obstacleDistance = 20 cm, then tune to 25 or 30 cm if it turns too late.

Expected Result

Robot drives forward while the path is clear. Near an obstacle, it stops, backs up, turns, and tries a new path.

Troubleshooting

Problem	What to check
Robot does not move	Motor battery missing, ENA/ENB not enabled, or motor driver GND not shared.
Arduino resets	Motor current draw is sagging power. Separate motor power from Arduino power.
Robot spins circles	One motor is reversed or one side speed is lower.
Crashes into objects	Increase obstacleDistance and aim sensor straight forward.

Arduino Code

```
const int trigPin=9, echoPin=10;
const int ena=5, in1=7, in2=8, in3=11, in4=12, enb=6;
const int motorSpeed=150, obstacleDistance=20;

void setup(){
  pinMode(trigPin,OUTPUT); pinMode(echoPin,INPUT);
  pinMode(ena,OUTPUT); pinMode(in1,OUTPUT); pinMode(in2,OUTPUT);
  pinMode(in3,OUTPUT); pinMode(in4,OUTPUT); pinMode(enb,OUTPUT);
  Serial.begin(9600);
}
void loop(){
  float d=getDistanceCm(); Serial.println(d);
  if(d>obstacleDistance || d==0) moveForward();
  else { stopMotors(); delay(250); moveBackward(); delay(450); stopMotors(); delay(150); turnRight(); delay(500);
  stopMotors(); delay(150); }
}
float getDistanceCm(){
  digitalWrite(trigPin,LOW); delayMicroseconds(2);
  digitalWrite(trigPin,HIGH); delayMicroseconds(10); digitalWrite(trigPin,LOW);
  long duration=pulseIn(echoPin,HIGH,30000); if(duration==0) return 0;
  return duration*0.0343/2;
}
void moveForward(){ analogWrite(ena,motorSpeed); analogWrite(enb,motorSpeed); digitalWrite(in1,HIGH);
digitalWrite(in2,LOW); digitalWrite(in3,HIGH); digitalWrite(in4,LOW);}
void moveBackward(){ analogWrite(ena,motorSpeed); analogWrite(enb,motorSpeed); digitalWrite(in1,LOW);
digitalWrite(in2,HIGH); digitalWrite(in3,LOW); digitalWrite(in4,HIGH);}
void turnRight(){ analogWrite(ena,motorSpeed); analogWrite(enb,motorSpeed); digitalWrite(in1,HIGH);
digitalWrite(in2,LOW); digitalWrite(in3,LOW); digitalWrite(in4,HIGH);}
void stopMotors(){ digitalWrite(in1,LOW); digitalWrite(in2,LOW); digitalWrite(in3,LOW); digitalWrite(in4,LOW);}
```

Scan the QR code on the package README or visit www.wolfieweb.com/works01.html for the live tutorial page.