

314492			
RFID Learning Kit (Contents)			
1x 50K Ω Potentiometer			
1x 7-seg LED 1x module			
1x 7-seg LED 4x module			
1x 8x8 dot LED array			
1x 9v battery cable			
1x Buzzer (active)			
1x Buzzer (passive)			
1x Flame sensor			
1x IC 74HC595N 16-pin DIP			
1x IR receiver			
1x IR remote control			
1x Joystick module			
1x LED - RGB			
1x LM35 Temp Sensor			
1x Microphone sound sensor			
1x Relay 5v			
1x RFID card			
1x RFID fob			
1x RFID sensor			
1x RTC module			
1x Servo Motor			
1x Stepper module			
1x Stepper Motor			
1x Temp & Humidity			
1x USB cable			
1x Water Level			
2x Ball tilt sensor			
3x Photo Resistor			
4x Large button switch			
5x 10K Ω resistor			
5x 1K Ω resistor			
5x LED - Blue			
5x LED - Red			
5x LED - Yellow			
830-pin Breadboard			
8x 220 Ω resistor			
Uno R3 compatible board			
Dupont connector wires			
1x 4*4 button switch module			
1x 2x16 LCD display			

## Inland Uno R3:



UNO R3 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 (ATmega328) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Clock Speed	16 MHz
Length	68.6 mm
Width	53.4 mm
Weight	25 g

See <u>http://arduino.cc</u> for detailed specifications, overviews, schematics, etc. Core functions, code examples, and links to many of the device libraries can be found in the learning section; refer to the manufacturer's site if using other add-on shields or sensors.

The latest Arduino Integrated Development Environment (IDE) necessary for programming your UNO R3 board can be obtained at <u>http://arduino.cc/en/Main/Software</u> (the **Download** menu choice on Arduino.cc)

Examples for many basic components can be found under the Examples menu. As you install libraries for additional shields, new examples may be available.

Follow the getting started guide found on the arduino.cc web site. Click **Learning**, and select **Getting started**. Click on the link for Windows, Mac OS X, or Linux for more specific directions.

Getting Started:

- 1. Download the Arduino Environment (IDE) and install or unzip/extract the application directory.
- 2. Connect the UNO board to one of your computer's USB port.
- Install the drivers (If the computer does not automatically download and install the necessary USB drivers, point the hardware setup to the "drivers" directory of the Arduino IDE application.)
- 4. Launch the Arduino IDE application
- 5. Open a sketch example such as "Blink"
- 6. Select your **Board** from the Tools menu.
- 7. Select the **Serial Port** used by the board
- 8. Upload the sketch to the board

sketch\_dec12a | Arduino 1.0 File Edit Sketch Tools Help Auto Format Ctrl+T  $\bigcirc \bigcirc$ ø Archive Sketch sketch\_dec12; Fix Encoding & Reload Ctrl+Shift+M Serial Monitor • Arduino Uno Board Serial Port Arduino Duemilanove w/ ATmega328 Arduino Diecimila or Duemilanove w/ ATmega168 Programmer Arduino Nano w/ ATmega328 Burn Bootloade 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 USE LilvPad 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 Arduino Robot Control Arduino Robot Motor

Sketch (code) Examples are included as part of the IDE. If you install device libraries for other components or shields, additional examples may be included and will show up in the list under the IDE File menu.

(See: <u>http://arduino.cc/en/Tutorial/HomePage</u> for an overview of the core functions and libraries.)

# **Components:**





LC	D



	Ball Tilt Sensor
	This is a very simple switch with a ball inside of the tube. When the sensor is tipped upward past the horizontal, the ball will short the contacts, closing the switch. With the top (away from the pins) is tilted down relative to the horizontal, the switch opens.
	LM35 Temperature Sensor
+VS VOUT GND BOTTOM VIEW LM35DZ	The LM35 series are precision integrated-circuit temperature sensors, whose output voltage is linearly proportional to the Celsius (Centigrade) temperature. Basic Temperature Sensor (+2° to +150°C): +Vs=5V in, Ground, Vout = 0mV + 10.0mV/°C For code examples, see: <u>http://playground.arduino.cc/Main/LM35HigherResolution</u>
<u>3p.pdf</u>	http://www.learningaboutelectronics.com/Articles/LM35-
	temperature-sensor-circuit.pnp
	http://pscmpf.blogspot.com/2008/12/arduino-Im35-sensor.html
	https://tkkrlab.nl/wiki/Arduino_KY- 001_Temperature_sensor_module
	Microphone Sound Sensor
	Pin connections:   G - connect to Ground   + - connect to 5V   AO - Analog Out - connect to Arduino for analog input   DO - Digital Out - Connect to Arduino as digital trigger input; adjust sensitivity via the screw on the potentiometer.   For code examples, see:   http://www.princetronics.com/sound-sensitive-lights-w-sound-sensor-arduino/
	https://tkkrlab.nl/wiki/Arduino_KY- 038 Microphone sound sensor module
Keyes_SJoys	Joystick with push button module Joystick module has five connections: GND (Ground), +5V, VRx = x-axis analog output, VRy = y-axis analog output, SW = Normally Open switch (push down on joystick to activate).
	For code examples, see: <u>http://arduino.cc/en/Tutorial/JoystickMouseControl</u> <u>https://tkkrlab.nl/wiki/Arduino_KY-023_XY-axis_joystick_module</u> <u>https://teamprincipia.wordpress.com/2007/12/27/joystick-control-</u> of-a-servo/

	Water Level or Rain Sensor
	Three connections - = Ground, + = 5V, S = analog signal that will vary based on how much of the contacts are in contact with water. See: <u>http://www.instructables.com/id/Water-Level-Sensor-Module-for-</u> <u>Arduino-AVR-ARM-STM3/</u>
	http://www.seeedstudio.com/wiki/Grove - Water Sensor
	(Video): <u>http://www.wearerobots.co.uk/funduino-water-level-</u> sensor-tutorial-and-application/
	Temperature and Humidity sensor
	Three connections:
	(-) = ground (-) Note the square solder pad.
	(center pin) = +5V
	(S)= Signai (digitai, seriai output)
	For the DHT11 library and information, see:
	http://playground.arduino.cc/main/DHT11Lib
	http://www.johnboucha.com/arduino-dht11-temperature-
	humidity/
	https://tkkrlab.nl/wiki/Arduino_KY-
	015_Temperature_and_humidity_sensor_module
Motors	
	Servo motor
191	
	Note that Servo motor color schemes may vary:
	Note that Servo motor color schemes may vary: Brown / Black = ground
	Note that Servo motor color schemes may vary: Brown / Black = ground Orange / Red = +5V
	Note that Servo motor color schemes may vary: Brown / Black = ground Orange / Red = +5V Yellow / White = signal (use digital PWM connection.)
	Note that Servo motor color schemes may vary: Brown / Black = ground Orange / Red = +5V Yellow / White = signal (use digital PWM connection.)
	Note that Servo motor color schemes may vary: Brown / Black = ground Orange / Red = +5V Yellow / White = signal (use digital PWM connection.) Small servo motors may be connected directly to the Arduino controller, or you may wish to provide separate
	Note that Servo motor color schemes may vary: Brown / Black = ground Orange / Red = +5V Yellow / White = signal (use digital PWM connection.) Small servo motors may be connected directly to the Arduino controller, or you may wish to provide separate power to the servos and only connect the data and a
	Note that Servo motor color schemes may vary: Brown / Black = ground Orange / Red = +5V Yellow / White = signal (use digital PWM connection.) Small servo motors may be connected directly to the Arduino controller, or you may wish to provide separate power to the servos and only connect the data and a common ground.
	Note that Servo motor color schemes may vary: Brown / Black = ground Orange / Red = +5V Yellow / White = signal (use digital PWM connection.) Small servo motors may be connected directly to the Arduino controller, or you may wish to provide separate power to the servos and only connect the data and a common ground.
	Note that Servo motor color schemes may vary: Brown / Black = ground Orange / Red = +5V Yellow / White = signal (use digital PWM connection.) Small servo motors may be connected directly to the Arduino controller, or you may wish to provide separate power to the servos and only connect the data and a common ground. Servo examples are include with the IDE, see:
	Note that Servo motor color schemes may vary: Brown / Black = ground Orange / Red = +5V Yellow / White = signal (use digital PWM connection.) Small servo motors may be connected directly to the Arduino controller, or you may wish to provide separate power to the servos and only connect the data and a common ground. Servo examples are include with the IDE, see: <u>http://arduino.cc/en/Tutorial/Knob</u> <u>http://arduino.cc/en/Tutorial/Sweep</u>
	Note that Servo motor color schemes may vary: Brown / Black = ground Orange / Red = +5V Yellow / White = signal (use digital PWM connection.) Small servo motors may be connected directly to the Arduino controller, or you may wish to provide separate power to the servos and only connect the data and a common ground. Servo examples are include with the IDE, see: <u>http://arduino.cc/en/Tutorial/Knob</u> <u>http://arduino.cc/en/Tutorial/Sweep</u>
	Note that Servo motor color schemes may vary: Brown / Black = ground Orange / Red = +5V Yellow / White = signal (use digital PWM connection.) Small servo motors may be connected directly to the Arduino controller, or you may wish to provide separate power to the servos and only connect the data and a common ground. Servo examples are include with the IDE, see: <u>http://arduino.cc/en/Tutorial/Knob</u> <u>http://arduino.cc/en/Tutorial/Sweep</u>
	Note that Servo motor color schemes may vary: Brown / Black = ground Orange / Red = +5V Yellow / White = signal (use digital PWM connection.) Small servo motors may be connected directly to the Arduino controller, or you may wish to provide separate power to the servos and only connect the data and a common ground. Servo examples are include with the IDE, see: <u>http://arduino.cc/en/Tutorial/Knob</u> <u>http://arduino.cc/en/Tutorial/Sweep</u>

### Stepper motor and controller Stepper IC = ULN2003AN (or similar)

The stepper motor included in the kit connects to the controller through a white connector socket. Four inputs connect to your Arduino IN1, IN2, IN3, and IN4. Power for the motor and controller is provided through the ground (-) and 5-to-12V (+) pins.

Use an external power supply for the motor to avoid damaging the Arduino. Connect the ground of your external supply to the ground of the Arduino and the signal IN# pins to digital outputs.

For sketch examples, see: http://arduino.cc/en/Tutorial/MotorKnob



### RFID



### **RFID-RC522** Read/Write module

Typical connections: Vcc - 5V or 3.3V RST - Arduino pin 5 GND - Ground MISO - Arduino pin 12 MOSI - Arduino pin 11 SCK - Arduino pin 13 (NSS, IRQ are not connected) See: <u>http://playground.arduino.cc/Learning/MFRC522</u>

#### https://labitat.dk/wiki/RFID RC522-AN

https://sites.google.com/site/arduinomega2560projects/home/leve <u>l-1/arduino-rfid-rc522</u>

http://www.grantgibson.co.uk/2012/04/how-to-get-started-withthe-mifare-mf522-an-and-arduino/

### other

Q <sub>B</sub> [ 1 Q <sub>C</sub> [ 2 Q <sub>D</sub> [ 3 Q <sub>E</sub> [ 4 Q <sub>F</sub> [ 5 Q <sub>G</sub> [ 6 Q <sub>H</sub> [ 7 GND [ 8	16 2 15 3 14 4 13 5 12 5 11 7 10 3 9	V <sub>CC</sub> Q <sub>A</sub> SER OE RCLK SRCLK SRCLR Q <sub>H</sub>	Tri-state 8-bit shift register IC SN74HC595N (or similar) The datasheet refers to the 74HC595 as an "8-bit serial-in, serial or parallel-out shift register with output latches; 3-state." In other words, you can use it to control 8 outputs at a time while only taking up a few pins on your microcontroller. See:
SI	N74HC595		http://arduino.cc/en/tutorial/ShiftOut

**Real Time Clock Module** (for DS1302, see below...) The Tiny RTC communicates with a microprocessor via the I2C serial interface. The real-time clock/calendar provides seconds, minutes, hours, day, date, month, and year information. Pin connections:

BAT: Battery voltage monitor (not used) GND: Ground VCC: +5V SDA: I2C data (connect to A4) SCL: I2C clock (connect to A5)

DS: DS18B20 Temp. Sensor output (connect to D2)

SQ: Square wave output (not used)

For library and code example, see:

http://www.hobbyist.co.nz/?q=real\_time\_clock http://playground.arduino.cc/code/time



## **Additional Resources:**

Several sites have hook-up and information and code examples on a variety of sensors, similar to, and including the ones found in this kit. Some sensors may be loose components or integrated into different board designs. If the documented sensor uses the same electronic component, then any code sketch documented may work with the sensors found in your kit. However, depending on the circuit design, the adjustments or sensitivity range may need to be modified slightly to achieve the desired result. Sites documenting these and other sensors include:

Arduino Playground Examples and additional libraries (code sketches available from the IDE *File, Examples* menu): <u>http://www.arduino.cc/en/Tutorial/HomePage</u>

Arduino Playground Tutorials: <u>http://playground.arduino.cc/Learning/Tutorials</u>

Forum.HobbyComponents.com: <u>http://forum.hobbycomponents.com/viewtopic.php?f=73&t=1320</u>

LinkSprite Wiki - Advanced Sensors Kit for Arduino: <u>http://linksprite.com/wiki/index.php5?title=Advanced\_Sensors\_Kit\_for\_Arduino</u>

TkkrLab.nl (Tukkerlab)Wiki: https://tkkrlab.nl/wiki/Arduino 37 sensors

University of Rhode Island (PDF coursework): <u>http://www.ele.uri.edu/courses/ele205/Arduino%20-%20Learning.pdf</u>

Freeduino.org: <u>http://www.freeduino.org/</u>

Arduino for Projects (PDF with 1193 projects): <u>http://duino4projects.com/arduino-projects-pdf/</u>

Lady Ada - Introduction to Arduino- step-by-step lessons: http://www.ladyada.net/learn/arduino/index.html

Tronixstuf Arduino Tutorials: <u>http://tronixstuff.com/tutorials/</u>

Earthshine Electronics Beginners Guide to Arduino: https://docs.google.com/file/d/0Bw\_ruMOtRDDgNXI3OTFGZXhIZ2c/edit?usp=sharing

Sheepdog's Guide to Arduino Programming: <u>http://sheepdogguides.com/arduino/FA1main.htm</u>