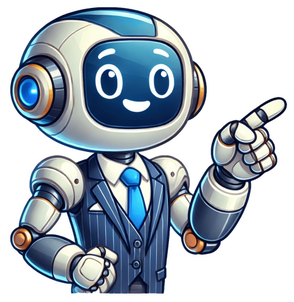
I'm not a bot



The Arduino UNO board has over 20 pins that you can use for various applications. In this tutorial, we'll explore the Arduino UNO Pinout and break down each section of Arduino Pins to discuss its functionality and use in detail. You'll learn how to use Arduino Pins in your project and get a solid understanding of a lot of Arduino fundamentals.Moreover, you'll also get the Arduino UNO pinout diagram and the Arduino R3 schematic diagram that you can use to create your own Arduino PCB board with any custom modifications that you'd need for your project. Without further ado, let's get right into it!Table of ContentsArduino UNO Pinout GuideIn the previous tutorial (Getting Started With Arduino), we discussed a lot of Arduino Hardware/Software fundamentals and the Arduino Ecosystem, we also prepared the development/simulation/prototyping environments, and we were able to develop and test our first 3 beginner projects.Now, we'll get a closer look at the Arduino UNO pinout and discuss the usage for each pin and its intended functionality. So, in this tutorial you'll mainly get the following points:Arduino UNO Pinout DiagramWhat are the Arduino pins you can use?What are their functionalities in detail?Some examples & use cases for each set of Arduino pinsDespite the fact that this article's main focus is Arduino UNO Pinout, it actually is still however very useful for Arduino beginners with different Arduino boards. The knowledge and basics we'll be discussing here are transferable to any other Arduino board even if it's not AVR-based.Arduino UNO Pinout DiagramHere is a detailed Arduino UNO Pinout Diagram with colored labels and legends to help you easily identify the functionalities that each pin can perform. It also has the Atmega328p AVR microcontroller pinout diagram to ease the mapping from the Arduino IO numbered pins to the actual microcontroller's pins.You can right-click and open in a new tab to view the full-resolution image and/or download it to your computer for reference."Pinout of ARDUINO Board and ATMega328PU" by pighixxx is licensed under Creative Commons Attribution-Share Alike 4.0 InternationalArduino UNO Pinout – Power PinsThe Arduino UNO board has a couple of LDO voltage regulators that provide regulated DC voltage of +5v and +3.3v to power up the main Arduino (Atmega328p) microcontroller. And you can also use the power pins to power up external modules and sensors that you'd like to use with your Arduino board.There are multiple options when it comes to powering up your Arduino UNO board that we'll discuss in detail hereafter in this section. Here is a figure for the power section on your Arduino UNO board with a detailed pinout for each power pin and its functionality.Arduino Ground PinsThe Arduino UNO board has 3 ground pins that you can use while connecting external devices like LEDs, Sensors, Modules, etc.+5v SupplyThis is a regulated DC voltage output of +5v that you can use to supply external sensors and modules from your Arduino board. Try not to overload it with heavy loads.+3.3v SupplyThis is a regulated DC voltage output of +3.3v that you can use to supply external sensors and modules from your Arduino board. As some sensors and modules are not 5v tolerant, you need to supply them with 3.3v DC and also use a level-shifter with Arduino IO pins as they are also at 5v level.Arduino VIN PinFor powering your Arduino UNO board, you have a few options to do so. The first of which is to use the USB cable to provide power to your Arduino board and be able to flash a new firmware to the microcontroller over the USB communication.Another option is to use the DC jack input to supply the Arduino board with DC voltage (6v-12v) and try not to exceed 9v because higher voltage input will cause the LDO regulators to overheat and potentially get damaged quickly.The last option is to use the VIN pin to provide an external voltage supply that's also within the (6v-12v) range exactly like the DC-Jack input. And do not under any circumstances connect multiple power sources at the same time. Use only one option of the 3 we've discussed before.Arduino IOREF PinThe Arduino IOREF pin provides a 5v logic voltage reference for Arduino Shields that need to use it.Arduino UNO Pinout – Digital PinsArduino GPIO (General-Purpose Input Output) Pins are the digital IO pins of the Arduino UNO board.There are 14 GPIO pins on the Arduino UNO board as shown in the figure below.IO PinsThe pins (0 to 13) are used as a digital input or output for the Arduino UNO board.Pin 13 is used for the onboard (built-in) LED.In output mode, the Arduino digital pin is:5v (when the digital pin is set HIGH)0v (when the digital pin is set to LOW).In input mode, the Arduino digital input pin will read a 0 or 1 depending on the voltage applied to the pin:0v–0.8v: read as 0 (logic 0 or LOW)2v–5v: read as 1 (logic 1 or HIGH)Any voltage level between 0.8v-2v is considered undefined input for the digital IO pin.There are two protection diodes for each IO pin that will clip excessive over/under voltage away from damaging any IO pin. However, you still need to be careful with the voltage levels of various components in your system and do level-shifting if needed.PWM PinsThe pins indicated by the "~" on the Arduino board are the PWM output pins. There are 6 PWM output pins on the Arduino UNO board (pins 3, 5, 6, 9, 10, and 11).The PWM (Pulse Width Modulation) is a timer mode in which the timer is used to generate a constantly changing digital signal with a specific frequency and duty cycle. PWM is widely used for so many applications like motor control, LED brightness control, and much more.If you need more than 6 PWM outputs for your application, you can either check another board other than Arduino UNO or search for an external PWM controller IC to extend the PWM output pins count. There is also a possibility to create Software PWM (software emulated) output signals but be careful that it's going to cost you a lot of CPU load.Check the tutorial below to learn more about Arduino PWM.Interrupt PinsInterrupt pins are basically digital IO pins that can be configured to be in input mode. And will provide an interrupt signal directly to the CPU whenever the pin state changes. The CPU will pause the current operation and jump to a predefined interrupt handler function and execute it till completion, then it resumes the main loop() function (application).Interrupt pins can provide a near-immediate response to external events like input pin state change and this can be really useful for different applications that include, but are not limited to, the following list:Motor Speed Measurement With Optical EncoderProximity Sensor InterfacingKeypad Matrix ScanningUltrasonic Sensor Interfacingand much moreThere are two types of interrupt pins in the Arduino (Atmega328p) microcontroller: dedicated external interrupt pins and pin-change interrupts.1. External InterruptsThose are two dedicated pins INT0 and INT1 that correspond to Arduino IO pins 2, and 3 respectively. Each interrupt pin of those two has its own interrupt vector and can have its unique interrupt handler function.Note that other Arduino boards may have more dedicated external interrupt pins. The Arduino UNO microcontroller has only 2 dedicated external interrupt pins. And there is no practical way of increasing the dedicated interrupt pins.2. PCINT (Pin Change Interrupts)All Arduino UNO digital pins can be used as interrupt pins using the PCINT hardware. This is not a dedicated interrupt signal source unlike the previously discussed external interrupt pints (INT0/INT1).You can easily notice that every IO pin in the Arduino UNO pinout diagram has a PCINTx label. Where x is the number of the pin change interrupt signal. Those signals are grouped together to generate 3 interrupt signals one for each port of pins (combined together). And the only way to know which pins have their state changed is to manually check for it in your code logic.Having to manually check which pin state has changed in each port is the major drawback for PCINT interrupts compared to the dedicated external interrupts (INT0/INT1).Timer/Counter PinThere are two pins (T0 and T1) that can be used as external clock sources for Timer0 and Timer1 respectively. This can be very useful in a lot of applications depending on the mode of operation for the Timer module. (T0 is Arduino Pin4 and T1 is Arduino Pin5)In Counter Mode, the T0/T1 pins can be used as an external clock source so you can feed your counter module with external pulses so it keeps counting those incoming pulses. There are several use cases to take advantage of this hardware feature such as:Building a Frequency Counter Using Timer Module in Counter ModeMeasuring Motor Speed Using Optical Encoder + Timer in Counter Modeand much moreBy counting the incoming pulses count over a fixed time interval, we can determine the frequency (or period) of the input signal. As stated before, this can be very useful for reading sensors like optical encoders, proximity sensors for object counting, and much more.In Timer Mode, the T0/T1 pins can be used as an external clock source so you can feed your timer module with an external time-base clock generated from any other device (or circuitry) outside of the Arduino (Atmega328) microcontroller itself. One application that you can think of for this hardware feature is the RTC clock (Real-Time Clock).External RTC crystal (32.768kHz) can feed its output waveform to the T0 or T1 pin which is going to be selected as a clock source. Set the prescaler to 32. And given that Timer0/Timer1 are both 10 bits. Now you'll get a perfect 1-second timer interrupt that you can use as a time base for your real-time clock application. TOUT = 2n x (Prescaler/FCLK), which turns out to be TOUT = 210 x (32/32768) = 1 second.ICP (Input Capture) PinsThere is one input capture pin in Arduino UNO (Atmega328p) which is ICP1 that corresponds to Arduino pin 8. This pin is used with the timer module in timer mode while it's free running. When the ICP pin state changes, the current value of the timer register is captured and saved in the input capture register. It's also referred to as ICU (Input Capture Unit).This can be very useful in so many applications that include the following:External input signal pulse width measurementMotor RPM Measurement With Optical Encoder + ICUUltrasonic Distance Sensor Interfacing With ICUFrequency Counter ApplicationsPOV LED Strip Displayand much moreICSP Port PinsThe ICSP (In-Circuit Serial Programming) Header provides an extra way to flash (program) the Atmega328p microcontroller on the Arduino UNO board over the SPI. However, you may rarely need to use it as long as the USB port is functioning properly.The Arduino bootloader is flashed only once to the microcontroller's IC and then USB programming is the way to go thereafter. Unless something wrong goes with your chip or if you've replaced the Atmega328p microcontroller chip with a brand new one that doesn't have the bootloader flashed to it. Then the ICSP port will come to the rescue.Arduino UNO Pinout – Analog PinsThe Arduino UNO has 6 analog input pins labeled from A0 to A5 as shown in the figure below. Those pins can be used with analog peripherals in the Arduino microcontroller such as: ADC (A/D Converter) and the Analog Comparator.The Analog input pins can also be used as general-purpose digital IO pins as we'll see hereafter in this section.ADC (A/D Converter)The ADC (Analog to Digital Converter) in Arduino is used to read analog voltage input and convert it to its digital representation. Given that the Arduino's ADC is 10 bits in resolution, the digital output range is therefore from 0 up to 1023. And the analog input range is from 0v up to 5v (which is the analog reference voltage VREF = +5v).You can use the interactive tool below to set an analog input voltage and see the ADC digital output value that corresponds to the analog input voltage. The output equation for the ADC is as follows: ADC Output = (Analog input voltage / VREF) x (2n – 1). Where VREF = 5v and n is the ADC resolution which is 10bits. Analog ComparatorThe analog input pins can be used with the internal Analog comparator in the Arduino's microcontroller (Atmega328p) instead of the ADC. The analog comparator is really an interesting peripheral and can be a very powerful alternative for the ADC in many applications.Imagine you need to monitor some input analog voltage and check if it goes below a certain threshold value. You can use the ADC and keep sampling the input pin thousands of times per second and overload the CPU with this hectic task. While you could have used the analog comparator instead and set a threshold to the input voltage, when a voltage drop occurs, the analog comparator will fire an interrupt and the CPU will immediately respond to it. Without overloading the CPU with ADC conversions and polling for the analog input voltage.There are so many applications where you might need to use the analog comparator which include, but are not limited to, the following:Threshold voltage monitoringBattery Charging ApplicationsZCD (Zero-Crossing Detection)Tachometersand much moreWe'll dedicate some future tutorials to discuss Arduino's internal analog comparator in more detail with example projects as well.AREF Pin (Analog Reference)The AREF (Analog Reference) pin can be used to provide an external reference voltage for the analog-to-digital conversion of inputs to the analog pins (A0-A5). The reference voltage essentially specifies the maximum analog input voltage after which the ADC's output will saturate at 1023.Let's recall the ADC's output equation which is: ADC Output = (Analog input voltage / VREF) x (2n – 1). Where the "default" VREF = 5v and n is the ADC resolution which is 10bits.Using the AREF pin we can change the default VREF voltage and set it to any desired voltage level which allows for more precise A/D conversions. Let's say we hook up the AREF pin to an external 2.5v DC supply, now we'll end up having double the resolution of the default 5v analog reference.Because instead of dividing the 0v-5v range into discrete 1024 digital levels, we'll now divide the 0v-2.5v range into discrete 1024 digital levels. Which is exactly double the default resolution. But you must make sure that the input analog voltage will not exceed 2.5v after setting the AREF to 2.5v. Otherwise, the ADC will saturate at 1023 if the analog input voltage exceeded the new AREF (2.5v). Use the interactive tool above to try it yourself and see the effect of lowering the AREF voltage from 5v down to 2.5v and how it increases the ADC resolution by a factor of 2.VREFAnalog Input VoltageADC Digital Output5v0.01v22.5v0.01v4For the exact same analog input (0.01v), you get more ADC ticks (levels) at VREF=2.5v compared to VREF=5v. Which is what we mean by the resolution is doubled (increased by a factor of 2).Using Analog Pins As Digital IOArduino Uno has a total of 14 digital GPIO pins, out of which 6 pins (from A0 to A5) are analog pins. The question is can we use Arduino analog pins as digital output pins? The answer is YES, and here is how to do it.First of all, the ADC is a valuable resource in any microcontroller. Under certain circumstances, you'd sacrifice the analog input channels just to get extra digital IO pins. Sometimes an external IO expander would be a better solution instead of sacrificing the ADC analog input channels.But let's say you've made your decision and you need to have an extra digital pin or two. And the analog input pins are the only hope you've got. Here is how you'd go about implementing this.Just deal with the analog input pins (A0 – A5) as if they're normal digital IO pins. This means you first need to call the pinMode() function to set the pin mode to OUTPUT. And then call the digitalWrite() function to change the pin state to HIGH or LOW. Check out the tutorial below to get more information about this with a couple of code examples.Arduino UNO Pinout – Serial Ports PinsThe Arduino UNO's microcontroller (Atmega328p) has 3 different serial communication ports (protocols). Those ports enable the Arduino to communicate with a huge variety of sensors and modules to extend its functionality and capabilities beyond the limits of the Atmega microcontroller.Serial communication is a method of data exchange between embedded controllers in the form of bits that are sent/received bit-by-bit in a series. Hence the name, serial communication. There are different hardware ports (protocols) for serial data communication of which (UART, I2C, and SPI) that are supported by the Arduino's microcontroller.Each serial communication port (protocol) has its own set of advantages/disadvantages and its best use cases. We'll dedicate a separate in-depth tutorial for each serial communication protocol of them. But here is a summarized overview of each port and its IO pins.Arduino UART PinsUART (Universal Asynchronous Receiver-Transmitter) is the most popular serial communication protocol in embedded microcontrollers. In Arduino, we typically use the UART module for serial communication with the PC via a USB-TTL converter to print serial messages on the serial monitor.The Arduino UNO has only one UART module which has two pins (RX for receiving serial data and TX for transmitting serial data). The UART pins' mapping is as follows: RX is Arduino pin 0, and TX is Arduino pin 1 respectively.Besides serial communication with your PC for debugging and such, the UART is also used for interfacing various sensors and modules such as:Bluetooth Module HC-05GSM/GPRS ModulesGPS Modulesand much moreArduino I2C PinsI2C or IIC (Inter-Integrated Circuit) is also a very popular serial communication protocol that's widely used by different sensors and modules in embedded systems. It consists of 2 pins only (one for serial data and one for the serial clock). The Arduino I2C pins are (SDA and SCL).The I2C is a multi-master multi-slave protocol that supports a large number of devices on the same 2-wire bus. Each device has a unique ID that others can use to directly address that specific device. And it's considered the most efficient serial communication bus in terms of the number of IO pins needed to establish a communication network of multiple devices.There are so many modules and sensors that we can interface with Arduino using the I2C bus, which include:RTC ClockOLED Display ScreensI2C IO Pins ExpanderExternal EEPROM Memory ChipsMPU6050 IMU (Gyro+Accelerometer)and much moreTypically, you won't need more than one I2C module on your Arduino microcontroller. Because if you need to add a new extra device to the I2C bus, it's as simple as hooking it up to the SCL and SDA lines and you're good to go. Unlike the UART which is a single master single slave communication port.Arduino SPI PinsSPI (Serial Peripheral Interface) is one of the fastest serial communication ports at all. You can use Arduino's SPI for communicating with various sensors and modules at much higher data rates compared to the other serial communication protocols.The SPI communication bus consists of 3 main lines (Serial Data Out, Serial Data In, and Serial Clock). The SPI master device needs also an extra IO pin to select the slave to which it'd like to communicate. This means we need an extra IO pin for each slave device added to the bus which is definitely going to limit the maximum addressable devices on the bus.The SPI bus lines are named according to a conversion that eases the wiring and usage for users, which goes as follows:MOSI (Master Output Slave Input)MISO (Master Input Slave Output)SCK (Serial Clock)SS (Slave Select) "Only Required For SPI Slave Devices"There are so many applications in which you'd need to use the Arduino SPI bus to interface various sensors and modules which include, but are not limited to, the following examples:SPI TFT Display InterfacingDot Matrix Display InterfacingExternal SPI Flash MemoryExternal SPI ADC ModulesnRF24L01 Wireless Modulesand much moreAs stated earlier, We'll dedicate a separate in-depth tutorial for each serial communication protocol (UART – I2C – SPI). And we'll create a handful of applications and example projects to practice what we'll learn.Those are summarized tables for the Arduino UNO pinout with brief descriptions of each pin's functionalities.Arduino Digital IO PinsArduino PinAtmega328p PinPWM?Alternative Special FunctionDescriptionD0PD0RXUART RX PinD1PD1TXUART TX PinD2PD2INT0External Interrupt Pin0D3PD3✓INT1External Interrupt Pin1D4PD4T0Timer0 External Clock Input PinD5PD5✓T1Timer1 External Clock Input PinD6PD6✓D7PD7D8PB0ICP1Input Capture Unit PinD9PB1✓D10PB2✓SSSPI Slave Select PinD11PB3✓MOSISPI Master-Out Slave-InD12PB4MISOSPI Master-In Slave-OutD13PB5SCKSPI Serial Clock PinArduino Analog Input PinsArduino PinAtmega328p PinAlternative Special FunctionDescriptionA0PC0A1PC1A2PC2A3PC3A4PC4SDAI2C Serial Data LineA5PC5SCLI2C Serial Clock LineAREFAREFAREFAnalog Voltage Reference Input PinArduino Power PinsPin NameDescription5v+5v DC Regulated Supply3.3v+3.3v DC Regulated SupplyGNDGroundRESETµC Reset LineVINInput DC Voltage (6v-12v). Try not to exceed 9vIOREFProvides 5v Logic Voltage Reference For Arduino Shields That Use itArduino UNO R3 Schematic DiagramAs an open-source project, the Arduino UNO board schematic design and parts list is completely open-source and available for everyone to download, modify, and create custom Arduino board designs based on it.You'll also get a much clear view of the underlying hardware details by carefully studying the Arduino UNO board schematic design files. You can download the Arduino UNO R3 schematic design diagram file using (this link).Get Started With Arduino UNO PinsTo conclude this tutorial, we'll highlight the fact that there are so many options for using Arduino pins and a unique set of features that each pin does support. We hope it's been a helpful guide to helping you decide which Arduino pins to use for your project and what are the fundamental limitations and workarounds for each.If you're just getting started with Arduino, you need to check out the Arduino Getting Started [Ultimate Guide] here.And follow this Arduino Series of Tutorials to learn more about Arduino Programming. Home / Hardware / UNO R3 Arduino UNO is a microcontroller board based on the ATmega328P. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, 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. You can tinker with your UNO without worrying too much about doing something wrong, worst case scenario you can replace the chip for a few dollars and start over again.The ATmega328P can easily be replaced, as it is not soldered to the board.The ATmega328P also features 1kb of EEPROM, a memory which is not erased when powered off.The Arduino UNO features a barrel plug connector, that works great with a standard 9V battery. Arduino Uno is a popular open-source microcontroller board that has revolutionized the world of DIY electronics. It is widely used by hobbyists, students, and professionals alike for creating interactive projects, prototypes, and gadgets. With its user-friendly interface and robust hardware capabilities, Arduino Uno makes programming easy even for beginners. Its compatibility with a vast array of sensors, actuators, and shields allows for endless customization possibilities. Arduino UNO is based on the Atmega328P microcontroller. Arduino Uno offers 14 digital input/output pins, 6 analog inputs, a USB connection for programming and power supply purposes, a power jack to connect external power sources, and more. Related Articles: Getting start with Arduino Controlling Multiple LEDs With Arduino Uno The Arduino Uno features a total of 20 pins that can serve various purposes, including digital input/output, analog input, power supply, and communication. The pins on the Arduino Uno are categorized into several groups, including special function pins (RESET, VIN), power pins (5V, 3.3V, GND), digital pins (0–13), and analog pins (A0-A5). You can use pins to link sensors, actuators, displays, and other electrical components. Each pin has a unique purpose. Out of the 14 digital pins on the board, 6 can function as PWM (Pulse Width Modulation) outputs. D3, D5, D6, D9, D10, and D11 are the PWM pins in Arduino UNO. PWM (Pulse Width Modulation) is a technique to get analog results from digital means. PWM mainly used in Arduino for controlling Servo Motors, LEDs, Speed Controlling, etc. You can power the Arduino UNO board either via USB connection or through the Power Jack using an external power supply. We have to use an AC to DC Adapter of 2.1 mm centre-positive plug into the board's power jack. The board can work on an external supply of 6 to 20 volts. Using a voltage higher than 12V may lead to overheating of the voltage regulator, potentially damaging the board. Vin:- The input voltage to the Arduino board when it's using an external power source you can supply voltage through this pin. If the supplying voltage via USB connection or the power jack, you can access it through this pin. 5V: – This pin gives the output of 5V. You can use this pin to give a 5V supply to the sensor. 3V3:- This pin gives the output of 3.3V. The maximum current drawn is 50 mA. GND:- These pins use as Ground Pins. IOREF:- On the Arduino board, this pin provides voltage references, with which the microcontroller operates. A properly configured gradient can read the IOREF pin voltage and can work with 5V or 3.3V to select a suitable power source on the output or enable the voltage translator on the output. The Arduino Uno is a popular microcontroller board known for its versatility and user-friendly design. When it comes to specifications, the Arduino Uno boasts an ATmega328P microcontroller running at 16MHz, with 32KB of flash memory and 2KB of SRAM. To function, this board needs a voltage of 7 to 12 volts, which can be provided via a USB connection or an external power source. It runs on a 5V voltage. This board also has integrated LEDs for pin status monitoring and power signaling. MicrocontrollerATmega328POperating Voltage5VInput Voltage (recommended)7-12VInput Voltage (limit)6-20VDigital I/O Pins14 (of which 6 provide PWM output)Analog Input Pins6DC Current per I/O Pin20 mADC Current for 3.3V Pin50 mAFlash Memory32 KB (ATmega328P) of which 0.5 KB used by bootloaderSRAM2 KB (ATmega328P)EEPROM1 KB (ATmega328P)Clock Speed16 MHzLED_BUILTIN13Length68.6 mmWidth53.4 mmWeight25 g Note:- Do not supply voltage via the 5V or 3.3V pins, it bypasses the regulator and can damage the board. We don't advise it. Arduino Uno is undeniably a fantastic choice for beginners stepping into the world of electronics and programming. Its user-friendly interface, extensive community support, and versatility make it an ideal platform to learn and experiment with various projects. With its straightforward pinout, specifications, and easy setup process, it enables users to dive right into creating innovative projects without feeling overwhelmed. Whether you are interested in robotics, home automation, or interactive art installations, Arduino Uno provides a solid foundation to bring your ideas to life. In this tutorial, we will have a brief overview about the Arduino UNO Board, the layout of a typical UNO board, some of the important specifications and finally the Arduino UNO Pinout along with the pin description. Arduino UNO Pinout Perhaps the most popular board in the Arduino line-up is the Arduino UNO. There are other boards like the Arduino Nano and the Arduino Mega, but UNO has been the go-to board for quick prototyping, Arduino Projects and DIY Projects. Arduino UNO in DIP and SMD Packages Arduino UNO is based on ATmega328P Microcontroller, an 8-bit AVR Architecture based MCU from ATMEL. Arduino UNO comes in two variants: one consists of a 28-pin DIP Microcontroller while the other consists of 32 lead Quad Flat Package Microcontroller. Other than that, rest is identical in both the boards. Arduino UNO Board Layout The following image shows the layout of a typical Arduino UNO board. All the components are placed on the top side of the PCB. Layout of Arduino UNO Board As you can notice, there is a Type-B USB connector on the left short edge of the board, which is used for powering on the board as well as programming the Microcontroller. There is also a 2.1 mm DC jack to provide external power supply. Apart from that, the layout of Arduino UNO is very much self-explanatory. I will discuss about the pins of Arduino UNO in the Arduino UNO Pinout Section. Technical Specifications of Arduino UNO As Arduino UNO is based on ATmega328P Microcontroller, the technical specifications of Arduino UNO are mostly related to the ATmega328P MCU. But none the less, let me give you a brief overview about some important specifications of Arduino UNO. MCU ATmega328P Architecture AVR Operating Voltage 5V Input Voltage 6V – 20V (limit) 7V – 12V (recommended) Clock Speed 16 MHz Flash Memory 32 KB (2 KB of this used by bootloader) SRAM 2 KB EEPROM 1 KB Digital IO Pins 24 (of which 6 can produce PWM) Analog Input Pins 6 How to power up the Arduino UNO? There are a couple of ways in which you can power the UNO board. The first and easy way is using the Type-B USB Connector. The next way is to provide an unregulated supply in the range of 6V to 20V to VIN pin of the UNO (Pin number 26). You can also supply the unregulated supply through the 2.1mm DC Jack, in which case, you can access the supplied voltage through the VIN Pin. What are Different Memories of Arduino UNO? Strictly speaking, this is specific to the MCU i.e., ATmega328P, used on the Arduino UNO Board. ATmega328P microcontroller comes with three different types of memories. They are: 32 KB of Flash Memory 2 KB of SRAM 1 KB of EEPROM 0.5 KB of the Flash Memory is used by the bootloader code. What are the Input and Output Pins of Arduino UNO? Of the 32 pins available on the UNO board, 22 pins are associated with input and output. In that 14 pins (D0 to D13) are true digital IO pins, which can be configured as per you application using pinMode(), digitalWrite() and digitalRead() functions. All these Digital IO pins are capable of sourcing or sinking 20mA of current (maximum 40mA is allowed). An additional feature of the Digital IO pins is the availability of internal pull-up resistor (which is not connected by default). The value of the internal pull-up resistor will be in the range of 20KΩ to 50KΩ. There are also 6 Analog Input Pins (A0 to A5). All the analog input pins provide a 10-bit resolution ADC feature, which can be read using analogRead() function. An important point about Analog Input pins is that they can be configured as Digital IO pins, if required. Digital IO pins 3, 5, 6, 9, 10 and 11 are capable of producing 8-bit PWM Signals. You can use analogWrite() function for this. What Communication Interfaces are available on Arduino UNO? Arduino UNO supports three different types of communication interfaces. They are: Perhaps the most common communication interface in the Arduino universe is the Serial Communication. In fact, the Arduino boards (UNO or Nano or Mega) are programmed using the serial communication. Digital IO pins 0 and 1 are used as Serial RX and TX pins to receive and transmit serial data. These pins are connected to the serial pins of the on-board USB to Serial Converter IC. Analog Input Pins A4 and A5 have alternative functions. These pins A4 and A5 are used as SDA (A4) and SCL (A5) respectively, which enables its support for I2C or I2C or Two-Wire Interface (TWI) communication. The final communication interface is the SPI. The Digital IO pins 10, 11, 12, and 13 of this Arduino Uno are configured for SPI communication as SS, MISO, SCK, and MOSI pins respectively. Any additional features? An on-board LED connected to digital IO pin 13. Use this LED to perform Blinky operations. The reference voltage for the internal ADC is by default set to 5V. But using the AREF pin, you can manually set the upper limit of the ADC. Using the IOREF pin, you can set the reference voltage for Microcontroller operations. To reset the microcontroller, you can use the on-board RESET button. Although you can program the Arduino UNO using the USB cable, there is a provision to program the MCU using the In-Circuit Serial Programming (ICSP) interface. The UART bootloader, which is preloaded in to the ATmega328P microcontroller, enables programming through serial interface. But ICSP doesn't need any bootloader. You can program Arduino UNO using ISCP or use the ISCP of Arduino UNO to program other Arduino Boards. Digital IO Pins 2 and 3 can be configured as External Interrupts Pins INT0 and INT1 respectively. Use attachInterrupt() function to configure the Interrupt for rising edge, falling edge or level change on the pin. Arduino UNO Pinout Now that we have seen a little bit about Arduino UNO and its important features and specifications, let us dive into the Arduino UNO Pinout. The following image shows the complete pinout of Arduino UNO Board. As you can see from the image, I described each pin of the Arduino UNO with its microcontroller equivalent pin, alternative functions, default functionality and other additional features. For higher resolution image, click here. Pin Description For pin description of Arduino UNO, let us assume some basic numbering. Let the numbering begin with the RX Pin (D0). So, RX is Pin 1, TX is Pin 2, D2 is Pin 3 and so on. On the other side, NC is Pin 19, IOREF is Pin 20 etc. Overall, there are 32 pins on the Arduino UNO Board. With this information, let us now see the pin description of Arduino UNO. Pin Number Pin Name Description Alternative Functions 1 RX / D0 Digital IO Pin 0 Serial RX Pin Generally used as RX 2 TX / D1 Digital IO Pin 1 Serial TX Pin Generally used as TX 3 D2 Digital IO Pin 2 4 D3 Digital IO Pin 3 Timer (OC2B) 5 D4 Digital IO Pin 4 Timer (T0/XCK) 6 D5 Digital IO Pin 5 Timer (OC0B/T1) 7 D6 Digital IO Pin 6 8 D7 Digital IO Pin 7 9 D8 Digital IO Pin 8 Timer (CLK0/ICP1) 10 D9 Digital IO Pin 9 Timer (OC1A) 11 D10 Digital IO Pin 10 Timer (OC1B) 12 D11 Digital IO Pin 11 SPI (MOSI) Timer (OC2A) 13 D12 Digital IO Pin 12 SPI (MISO) 14 D13 Digital IO Pin 13 SPI (SCK) 15 GND Ground 16 AREF Analog Reference 17 SDA / D18 Digital IO Pin 18 I2C Data Pin 18 SCL / D19 Digital IO Pin 19 I2C Clock Pin 19 NC Not Connected 20 IOREF Voltage Reference 21 RESET Reset (Active LOW) 22 3V3 Power 23 5V +5V Output from regulator or +5V regulated Input 24 GND Ground 25 GND Ground 26 VIN Unregulated Supply 27 A0 Analog Input 0 Digital IO Pin 14 28 A1 Analog Input 1 Digital IO Pin 15 29 A2 Analog Input 2 Digital IO Pin 16 30 A3 Analog Input 3 Digital IO Pin 17 31 A4 Analog Input 4 Digital IO Pin 18 I2C (SDA) 32 A5 Analog Input 5 Digital IO Pin 19 I2C (SCL) The following table describes the pins of the ICSP Connector. MISO Master In Slave Out (Input or Output) 5V Supply SCK Clock (from Master to Slave) MOSI Master Out Slave In (Input or Output) RESET Reset (Active LOW) GND Ground There is also a similar ICSP connector known as ICSP1 associated with the ATmega16U Microcontroller. For more information on this connector, take a look at the Arduino UNO Pinout image. Conclusion This was a brief overview on Arduino UNO board layout, technical specifications, important features and most importantly the complete Arduino UNO Pinout information. If you have any doubts, don't hesitate them to share them with us in the comments section below. We will help you to clear them. Today, I am going to uncover the details on the Introduction to Arduino Uno. It is a microcontroller board developed by Arduino and based on Atmega328. Hi Friends! Hope you are doing great. Today, I am going to give you a detailed Introduction to Arduino Uno. It is a microcontroller board developed by Arduino.cc and is based on Atmega328 Microcontroller. The first Arduino project was started in Interaction Design Institute Ivrea in 2003 by David Cuartielles and Massimo Banzi with the intention of providing a cheap and flexible way for students and professionals to learn embedded programming. Arduino UNO is a very valuable addition in electronics that consists of a USB interface, 14 digital I/O pins(of which 6 Pins are used for PWM), 6 analog pins and an Atmega328 microcontroller. It also supports 3 communication protocols named Serial, I2C and SPI protocol. You should also have a look at this video presentation on Arduino UNO: Few main features of Arduino UNO are shown in the below figure: Arduino UNO Features and Technical Specs No. Parameter Name Parameter Value 1 Microcontroller Atmega328 2 Crystal Oscillator 16MHz 3 Operating Voltage 5V 4 Input Voltage 5-12V 5 Digital I/O Pins 14 (D0 to D13) 6 Analog I/O Pins 6 (A0 to A5) 7 PWM Pins 6 (Pin # 3, 5, 6, 9, 10 and 11) 8 Power Pins 5V, 3.3V, Vin, GND 9 Communication UART(1), SPI(1), I2C(1) 10 Flash Memory 32 KB (0.5KB is used by bootloader) 11 SRAM 2 KB 12 EEPROM 1 KB 13 ICSP Header Yes 14 Power sources DC Power Jack & USB Port I'll try to cover each and everything related to Arduino Uno, so you get a clear idea of what it does, its main features, working and everything you need to know. Let's get started. Where To Buy?No.ComponentsDistributorLink To Buy1Arduino UnoAmazonBuy Now Arduino Uno is a microcontroller board, developed by Arduino.cc, based on the Atmega328 microcontroller and is marked as the first Arduino board developed(UNO means "one" in Italian). The software used for writing, compiling & uploading code to Arduino boards is called Arduino IDE (Integrated Development Environment), which is free to download from Arduino Official Site. It has an operating voltage of 5V while the input voltage may vary from 7V to 12V. Arduino UNO has a maximum current rating of 40mA, so the load shouldn't exceed this current rating or you may harm the board. It comes with a crystal oscillator of 16MHz, which is its operating frequency. Arduino Uno Pinout consists of 14 digital pins starting from D0 to D13. It also has 6 analog pins starting from A0 to A5. It also has 1 Reset Pin, which is used to reset the board programmatically. In order to reset the board, we need to make this pin LOW. It also has 6 Power Pins, which provide different voltage levels. Out of 14 digital pins, 6 pins are used for generating PWM pulses of 8-Bit resolution. PWM pins in Arduino UNO are D3, D5, D6, D9, D10 and D11. Arduino UNO comes with 3 types of memories associated with it, named: Flash Memory: 32KB SRAM: 2KB EEPROM: 1KB Arduino UNO supports 3 types of communication protocols, used for interfacing with third-party peripherals, named: Serial Protocol I2C Protocol SPI Protocol You can download the Arduino UNO datasheet by clicking the below button: Download Arduino UNO Datasheet Apart from USB, a battery or AC to DC adopter can also be used to power the board. Features of Arduino Uno Board Arduino Uno comes with a USB interface i.e. USB port is added on the board to develop serial communication with the computer. Atmega328 microcontroller is placed on the board that comes with a number of features like timers, counters, interrupts, PWM, CPU, I/O pins and based on a 16MHz clock that helps in producing more frequency and number of instructions per cycle. It is an open-source platform where anyone can modify and optimize the board based on the number of instructions and tasks they want to achieve. This board comes with a built-in regulation feature that keeps the voltage under control when the device is connected to the external device. A reset pin is present in the board that resets the whole board and takes the running program in the initial stage. This pin is useful when the board hangs up in the middle of the running program; pushing this pin will clear everything up in the program and starts the program right from the beginning. There are 14 I/O digital and 6 analog pins incorporated in the board that allows the external connection with any circuit with the board. These pins provide flexibility and ease of use to the external devices that can be connected through these pins. There is no hard and fast interface required to connect the devices to the board. Simply plug the external device into the pins of the board that are laid out on the board in the form of the header. The 6 analog pins are marked as A0 to A5 and come with a resolution of 10bits. These pins measure from 0 to 5V, however, they can be configured to the high range using analogReference() function and AREF pin. Only 5 V is required to turn the board on, which can be achieved directly using a USB port or external adopter, however, it can support an external power source up to 12 V which can be regulated and limit to 5 V or 3.3 V based on the requirement of the project. Arduino Uno Pinout Arduino Uno is based on an AVR microcontroller called Atmega328. This controller comes with 2KB SRAM, 32KB of flash memory, 1KB of EEPROM. Arduino Board comes with 14 digital pins and 6 analog pins. ON-chip ADC is used to sample these pins. A 16 MHz frequency crystal oscillator is equipped on the board. The following figure shows the pinout of the Arduino Uno Board. Arduino UNO Pin Description There are several I/O digital and analog pins placed on the board which operates at 5V. These pins come with standard operating ratings ranging between 20mA to 40mA. Internal pull-up resistors are used in the board that limits the current exceeding the given operating conditions. However, too much increase in current makes these resisters useless and damages the device. LED. Arduino Uno comes with a built-in LED which is connected through pin 13. Providing HIGH value to the pin will turn it ON and LOW will turn it OFF. Vin. It is the input voltage provided to the Arduino Board. It is different than 5 V supplied through a USB port. This pin is used to supply voltage. If a voltage is provided through a power jack, it can be accessed through this pin. 5V. This board comes with the ability to provide voltage regulation. 5V pin is used to provide output regulated voltage. The board is powered up using three ways i.e. USB, Vin pin of the board or DC power jack. USB supports voltage around 5V while Vin and Power Jack support a voltage ranges between 7V to 20V. It is recommended to operate the board on 5V. It is important to note that, if a voltage is supplied through 5V or 3.3V pins, they result in bypassing the voltage regulator that can damage the board if the voltage surpasses its limit. GND. These are ground pins. More than one ground pins are provided on the board which can be used as per requirement. Reset. This pin is incorporated on the board which resets the program running on the board. Instead of physical reset on the board, IDE comes with a feature of resetting the board through programming. IOREF. This pin is very useful for providing voltage reference to the board. A shield is used to read the voltage across this pin which then selects the proper power source. PWM. PWM is provided by 3,5,6,9,10, 11pins. These pins are configured to provided 8-bit output PWM. SPI. It is known as Serial Peripheral Interface. Four pins 10(SS), 11(MOSI), 12(MISO), 13(SCK) provide SPI communication with the help of the SPI library. AREF. It is called Analog Reference. This pin is used for providing a reference voltage to the analog inputs. TWI. It is called Two-wire Interface. TWI communication is accessed through Wire Library. A4 and A5 pins are used for this purpose. Serial Communication. Serial communication is carried out through two pins called P