**microcontroller**

A microcontroller is a compact integrated circuit designed to govern a specific operation in an [embedded system](https://internetofthingsagenda.techtarget.com/definition/embedded-system). A typical microcontroller includes a [processor](https://whatis.techtarget.com/definition/processor), [memory](https://searchstorage.techtarget.com/definition/memory-card) and input/output (I/O) [peripherals](https://searchmobilecomputing.techtarget.com/definition/peripheral) on a single chip.

Sometimes referred to as an embedded controller or microcontroller unit (MCU), microcontrollers are found in vehicles, [robots](https://searchenterpriseai.techtarget.com/definition/robot), office machines, medical devices, mobile radio transceivers, vending machines and home appliances among other devices.

**Microcontroller features**

A microcontroller's processor will vary by application. Options range from the simple 4-[bit](https://whatis.techtarget.com/definition/bit-binary-digit), 8-bit or 16-bit processors to more complex 32-bit or 64-bit processors. In terms of memory, microcontrollers can use random access memory ([RAM](https://searchstorage.techtarget.com/definition/RAM-random-access-memory)), [flash memory](https://searchstorage.techtarget.com/definition/flash-memory), [EPROM](https://whatis.techtarget.com/definition/EPROM) or [EEPROM](https://whatis.techtarget.com/definition/EEPROM-electrically-erasable-programmable-read-only-memory). Generally, microcontrollers are designed to be readily usable without additional computing components because they are designed with sufficient onboard memory as well as offering pins for general I/O operations, so they can directly interface with [sensors](https://whatis.techtarget.com/definition/sensor) and other components.

Microcontroller architecture can be based on the Harvard architecture or von Neumann architecture, both offering different methods of exchanging data between the processor and memory. With a Harvard architecture, the data bus and instruction are separate, allowing for simultaneous transfers. With a Von Neumann architecture, one bus is used for both data and instructions.

Microcontroller processors can be based on [complex instruction set computing (CISC)](https://whatis.techtarget.com/definition/CISC-complex-instruction-set-computer-or-computing) or [reduced instruction set computing (RISC)](https://search400.techtarget.com/definition/RISC). CISC generally has around 80 instructions while RISC has about 30, as well as more addressing modes, 12-24 compared to RISC's 3-5. While CISC can be easier to implement and has more efficient memory use, it can have performance degradation due to the higher number of clock cycles needed to execute instructions. RISC, which places more emphasis on software, often provides better performance than CISC processors, which place more emphasis on hardware, due to its simplified instruction set and, therefore, increased design simplicity, but because of the emphasis it places on software, software can be more complex. Which ISC is used varies depending on application.

When they first became available, microcontrollers solely used assembly language. Today, the [C programming language](https://searchwindowsserver.techtarget.com/definition/C) is a popular option.

MCUs feature input and output pins to implement peripheral functions. Such functions include [analog-to-digital converters](https://whatis.techtarget.com/definition/analog-to-digital-conversion-ADC), liquid crystal display ([LCD](https://whatis.techtarget.com/definition/LCD-liquid-crystal-display)) controllers, [real-time clock (RTC)](https://whatis.techtarget.com/definition/real-time-clock-RTC), synchronous/asynchronous receiver transmitter (USART), timers, [universal asynchronous receiver transmitter (UART)](https://whatis.techtarget.com/definition/UART-Universal-Asynchronous-Receiver-Transmitter) and universal serial bus ([USB](https://whatis.techtarget.com/definition/USB-Universal-Serial-Bus-USB-30-SuperSpeed-USB)) connectivity. Sensors gathering data related to humidity and temperature among others are also often attached to microcontrollers.

**Types of microcontrollers**

Common MCUs include the Intel MCS-51, often referred to as an 8051 microcontroller, which was first developed in 1985; the AVR microcontroller developed by Atmel in 1996; the programmable interface controller (PIC) from Microchip Technology; and various licensed ARM microcontrollers.

A number of companies manufacture and sell microcontrollers, including NXP Semiconductor, Renesas Electronics, Silicon Labs and Texas Instruments.

**Microcontroller applications**

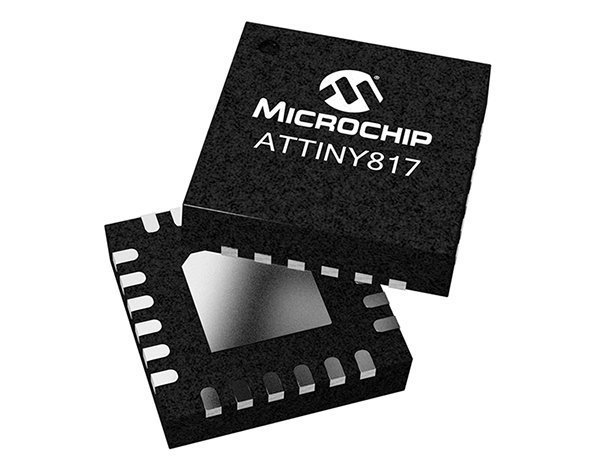
Microcontrollers are used in multiple industries and applications, including in the home and enterprise, building automation, manufacturing, robotics, automotive, lighting, smart energy, industrial automation, communications and [internet of things](https://internetofthingsagenda.techtarget.com/definition/Internet-of-Things-IoT) (IoT) deployments.

The simplest microcontrollers facilitate the operation of electromechanical systems found in everyday convenience items, such as ovens, refrigerators, toasters, mobile devices, [key fobs](https://searchsecurity.techtarget.com/definition/key-fob), video games, televisions and lawn-watering systems. They are also common in office machines such as photocopiers, [scanners](https://whatis.techtarget.com/definition/scanner), [fax machines](https://searchnetworking.techtarget.com/definition/fax) and [printers](https://whatis.techtarget.com/definition/printer), as well as [smart meters](https://internetofthingsagenda.techtarget.com/definition/smart-meter), ATMs and security systems.

More sophisticated microcontrollers perform critical functions in aircraft, spacecraft, ocean-going vessels, vehicles, medical and life-support systems, and robots. In medical scenarios, microcontrollers can regulate the operations of an artificial heart, kidney or other organ. They can also be instrumental in the functioning of prosthetic devices.

**Microcontrollers vs. microprocessors**

The distinction between microcontrollers and [microprocessors](https://whatis.techtarget.com/definition/microprocessor-logic-chip) has gotten less clear as chip density and complexity has become relatively cheap to manufacture and microcontrollers have thus integrated more "general computer" types of functionality. On the whole, though, microcontrollers can be said to function usefully on their own, with direct connection to sensors and actuators, where microprocessors are designed to maximize compute power on the chip, with internal bus connections (rather than direct I/O) to supporting hardware such as RAM and serial ports. Simply put, coffee makers use microcontrollers; desktop computers use microprocessors.



Microchip Technology Inc.

The Microchip Technology ATtiny817 microcontroller.

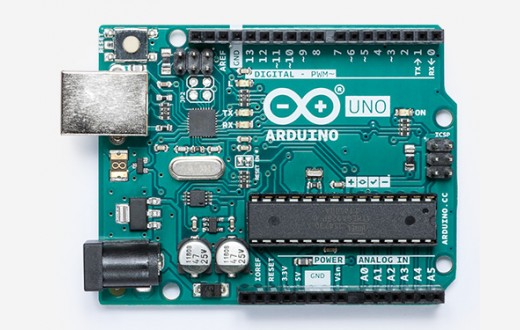
Microcontrollers are less expensive and use less power than microprocessors. Microprocessors do not have built in RAM, read-only memory ([ROM](https://whatis.techtarget.com/definition/read-only-memory-ROM)) or other peripherals on the chip, but rather attach to these with their pins. A microprocessor can be considered the heart of a computer system, whereas a microcontroller can be considered the heart of an embedded system.

**Choosing the right microcontroller**

There are a number of technology and business considerations to keep in mind when choosing a microcontroller for a project.

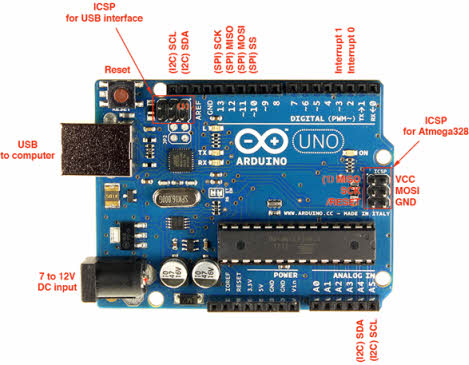
Beyond cost, it is important to consider the maximum speed, amount of RAM or ROM, and number or types of I/O pins on an MCU, as well as power consumption and constraints and development support.

**Arduino Uno Rev3**



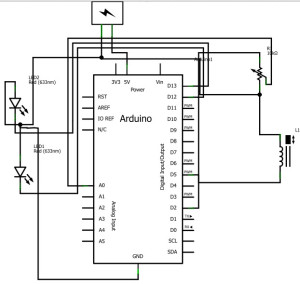
**Arduino Uno** is a microcontroller board based on the ATmega328P ([datasheet](http://www.atmel.com/Images/doc8161.pdf)). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, an ICSP header and a reset 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 worring too much about doing something wrong, worst case scenario you can replace the chip for a few dollars and start over again.

"Uno" means one in Italian and was chosen to mark the release of Arduino Software (IDE) 1.0. The Uno board and version 1.0 of Arduino Software (IDE) were the reference versions of Arduino, now evolved to newer releases. The Uno board is the first in a series of USB Arduino boards, and the reference model for the Arduino platform; for an extensive list of current, past or outdated boards see the Arduino index of boards.



Arduino is a single-board microcontroller meant to make the application more accessible which are interactive objects and its surroundings. The hardware features with an open-source hardware board designed around an 8-bit Atmel [AVR microcontroller](https://www.elprocus.com/types-of-avr-microcontroller-atmega32-and-atmega8/) or a 32-bit Atmel ARM. Current models consists a USB interface, 6 analog input pins and 14 digital I/O pins that allows the user to attach various extension boards.

The Arduino Uno board is a [microcontroller based](http://www.edgefx.in/microcontroller-based-projects-on-car-security-systems-using-gsm/) on the ATmega328. It has 14 digital input/output pins in which 6 can be used as PWM outputs, a 16 MHz ceramic resonator, an ICSP header, a USB connection, 6 analog inputs, a power jack and a reset button. This contains all the required support needed for microcontroller. In order to get started, they are simply connected to a computer with a USB cable or with a AC-to-DC adapter or battery. Arduino Uno Board varies from all other boards and they will not use the FTDI USB-to-serial driver chip in them. It is featured by the Atmega16U2 (Atmega8U2 up to version R2) programmed as a USB-to-serial converter.

[](http://www.edgefx.in/wp-content/uploads/2014/07/arduino-circuit.jpg)

**Arduino Uno with Digital Input/Output**

There are various types of Arduino boards in which many of them were third-party compatible versions. The most official versions available are the Arduino Uno R3 and the Arduino Nano V3. Both of these run a 16MHz Atmel ATmega328P 8-bit microcontroller with 32KB of flash RAM 14 digital I/O and six analogue I/O and the 32KB will not sound like as if running Windows. Arduino projects can be stand-alone or they can communicate with software on running on a computer. For e.g. Flash, Processing, Max/MSP). The board is clocked by a 16 MHz ceramic resonator and has a USB connection for power and communication. You can easily add micro SD/SD card storage for bigger tasks.

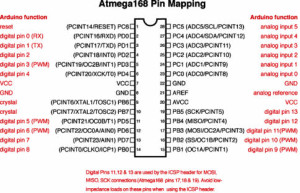
## Features of the Arduino Uno Board:

* It is an easy USB interface. This allows interface with USB as this is like a serial device.
* The chip on the board plugs straight into your USB port and supports on your computer as a virtual serial port. The benefit of this setup is that serial communication is an extremely easy protocol which is time-tested and USB makes connection with modern computers and makes it comfortable.
* It is [easy-to-find the microcontroller](https://www.elprocus.com/microcontrollers-types-and-applications/" \t "_blank) brain which is the ATmega328 chip. It has more number of hardware features like timers, external and internal interrupts, PWM pins and multiple sleep modes.
* It is an open source design and there is an advantage of being open source is that it has a large community of people using and troubleshooting it. This makes it easy to help in debugging projects.
* It is a 16 MHz clock which is fast enough for most applications and does not speeds up the microcontroller.
* It is very convenient to manage power inside it and it had a feature of built-in voltage regulation. This can also be powered directly off a USB port without any external power. You can connect an external power source of upto 12v and this regulates it to both 5v and 3.3v.
* 13 digital pins and 6 analog pins. This sort of pins allows you to connect hardware to your Arduino Uno board externally. These pins are used as a key for extending the computing capability of the Arduino Uno into the real world. Simply plug your electronic devices and [sensors](http://www.edgefx.in/6-different-types-of-temperature-sensors-with-their-specifications/" \t "_blank) into the sockets that correspond to each of these pins and you are good to go.
* This has an ICSP connector for bypassing the USB port and interfacing the Arduino directly as a serial device. This port is necessary to re-bootload your chip if it corrupts and can no longer used to your computer.
* It has a 32 KB of flash memory for storing your code.
* An on-board LED is attached to digital pin 13 to make fast the debugging of code and to make the debug process easy.
* Finally, it has a button to reset the program on the chip.

Arduino was created in the year 2005 by two Italian engineers David Cuartielles and Massimo Banzi with the goal of keeping in mind about students to make them learn how to program the Arduino uno microcontroller and improve their skills about electronics and use it in the real world.

Arduino uno microcontroller 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 is programmed using the Arduino programming language (based on Wiring) and the Arduino development environment (based on Processing).

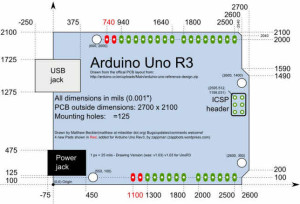
**ATmega168/328-Arduino Pin Mapping:**

[](http://www.edgefx.in/wp-content/uploads/2014/07/ATmega168-328Arduino-Pin-Mapping.jpg)

**ATmega168-328Arduino Pin Mapping**

### Programming:

* The Arduino integrated development environment (IDE) is a cross-platform application written in Java, and is derived from the IDE for the Processing programming language and the Wiring projects
* The Arduino Uno board can be programmed with the Arduino software.
* Select “Arduino Uno from the Tools > Board menu (according to the microcontroller on your board).
* The ATmega328 on the Arduino Uno 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.
* You can also bypass the [bootloader and program the microcontroller](http://www.edgefx.in/know-about-boot-loader-technique-for-programming-microcontroller/" \t "_blank) through the ICSP (In-Circuit Serial Programming) header.
* The ATmega16U2 (or 8U2 in the rev1 and rev2 boards) firmware source code is available .

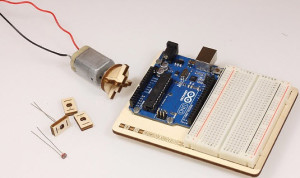
[](http://www.edgefx.in/wp-content/uploads/2014/07/arduino-mapping.jpg)

**Pin Diagram of Arduino Uno**

The ATmega16U2/8U2 is loaded with a DFU bootloader, which can be activated by:

* On Rev1 boards: connecting the solder jumper on the back of the board (near the map of Italy) and then resetting the 8U2.
* On Rev2 or later boards: there is a resistor that pulling the 8U2/16U2 HWB line to ground, making it easier to put into DFU mode.

You can then use Atmel’s FLIP software (Windows) or the DFU programmer (Mac OS X and Linux) to load a new firmware. Or you can use the ISP header with an external programmer (overwriting the DFU bootloader).

[](http://www.edgefx.in/wp-content/uploads/2014/07/arduino-uno-starter-kit.jpg)

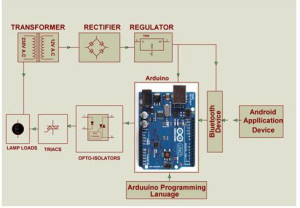
**Arduino Uno Starter Kit**

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

### Real-Time Applications of Arduino Uno Board

#### Arduino Based Home Automation System

The project is designed by using Arduino uno board for the development of home [automation system](http://www.edgefx.in/applications-of-short-range-technologies-using-zigbee-technology/" \t "_blank) with Bluetooth which is remotely [controlled and operated by an Android OS smart phone](http://www.edgefxkits.com/android-based-smart-phone-used-for-induction-motor-control" \t "_blank). Houses are becoming smarter and well developed by using such kind of advanced technologies. Modern houses are gradually increasing the way of design by shifting to centralized control system with [remote controlled switches](http://www.edgefx.in/types-of-remote-light-switches-and-thier-working-principles/" \t "_blank) instead of conventional switches.

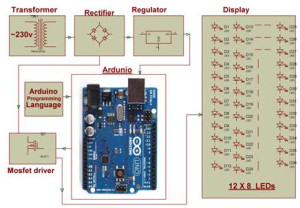
[](http://www.edgefx.in/wp-content/uploads/2014/07/arduino-board-home-automation.jpg)

**Arduino Based Home Automation**

In order to achieve this, a Bluetooth module is interfaced to the Arduino Uno board at the receiver end while on the transmitter end, a Graphical User Interface application on the cell phone sends ON/OFF commands to the receiver where loads are connected. By touching the identified location on the Graphical User Interface, lamps are used as loads in this project can be turned ON/OFF remotely by using this technology. The loads are operated by using Arduino Uno board through thyristors using triacs and OPTO-Isolators.

#### Arduino based Auto Intensity Control of Street Lights

As the intensity is cannot be controlled by using High Intensity Discharge (HID) lamps power saving is not possible in [street lights](http://www.edgefx.in/requirements-for-solar-led-street-light-system-and-its-applications/" \t "_blank) with these lamps as the density on roads is decreasing from peak hours of nights to early morning.

[](http://www.edgefx.in/wp-content/uploads/2014/07/arduino-based-auto-intensity-control.jpg)

**Arduino Based Auto Intensity Control**

Thus, this system overcomes this problem by [controlling the intensity of LED lights](http://www.edgefx.in/solar-based-solar-power-generation-projects-for-engineering-students/" \t "_blank) on street by gradually reducing intensity by controlling the voltage applied to these lamps. This system uses arduino board to [produce PWM pulses](https://www.elprocus.com/pulse-width-modulation-pwm/" \t "_blank) and it is programmed in such a way that it decreases the voltage applied to these lamps gradually till late nights and completely shutdowns at morning.

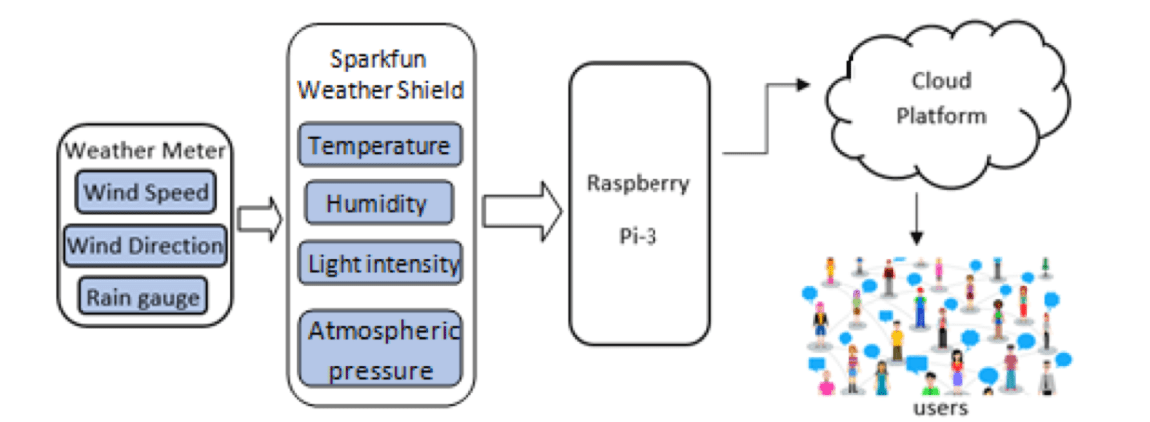
Thus, Arduino development board can sense the environment by receiving input from [different sensors](https://www.elprocus.com/sensors-types-applications/" \t "_blank) and affects its surroundings by controlling motors, lights and other actuators. The microcontroller on the board is programmed using the Arduino programming language. Thanks for your attention to this article and clarify doubts about [Arduino projects](http://www.edgefx.in/arduino-based-projects-on-security-system-for-engineering-students/" \t "_blank) by commenting below.

Raspberry pi3



Over the past four years, the Raspberry Pi has become known as the world’s most successful and accessible computer platform that anyone can program. It’s the kind of thing that reminds us fondly of the early 8-bit Apple II, Atari, and Commodore days, and with obviously much more capability now. It’s being used in schools across the globe, and a NASA astronaut even [took one along to the International Space Station](http://www.geek.com/science/this-gorgeous-raspberry-pi-case-is-heading-to-iss-1622598/). The Raspberry Pi Foundation has now launched the Raspberry Pi 3, an upgraded model that is on sale now for the same $35 price, just a bit more than a year after the Pi 2 landed.

The [Raspberry Pi 3](https://r.zdbb.net/u/12vc?or=https%3A%2F%2Fwww.extremetech.com%2Fcomputing%2F223769-raspberry-pi-3-launches-with-faster-64-bit-processor-and-wi-fi-for-the-same-35) includes a new Broadcom BCM2837 SoC with a 64-bit processor for the first time — a 1.2GHz quad-core ARM Cortex-A53 CPU that the company claims is roughly 10 times faster than the processor in the original model (at least on SysBench). The 33% bump in clock speed over the Pi 2’s 900MHz should deliver a 50-60% increase in performance in 32-bit mode.

[](https://www.google.com/url?sa=i&rct=j&q=&esrc=s&source=images&cd=&ved=2ahUKEwjc0NDCn97fAhVY7WEKHbZuDH4QjRx6BAgBEAU&url=https%3A%2F%2Fwww.favoriot.com%2Fiot-based-weather-station-by-using-raspberry-pi-3%2F&psig=AOvVaw1ncWMwZKdB8gF2cpT-g_iI&ust=1547037972498012)

**Fig**-IOT based Weather station system by using Raspberry pi3

**Advantages**

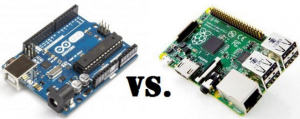
Although Raspberry Pi is as small as the size of a credit card, it works as if a normal computer at a relatively low price. it is possible to work as a low-cost server to handle light internal or web traffic. Grouping a set of Raspberry Pi to work as a server is more cost-effective than a normal server. If all light traffic servers are changed into Raspberry Pi, it can certainly minimize an enterprise’s budget.

**Disadvantages**

Even though Raspberry Pi can perform different tasks, there are some limitations due to its hardware. Because of its processor, it cannot run X86 operating systems. Some common ones like Windows and Linux distros are not compatible. In addition, some applications which require high demands on CPU processing are off-limits. “Model B took 107 ms to complete one calculation of the purely synthetic prime number test; a mid-range desktop Core 2 Duo E8400 took only 0.85ms.” (Collins, 2012) Users must not use normal computer standards to judge Raspberry Pi. It can work as a personal computer, but cannot replace it.

#### Arduino Vs Raspberry Pi

Choosing which board you want depends on the type of project you want to make, and your experience in programming. If you have no experience in programming or electronics, you will find the Arduino a steeper learning curve than the Raspberry Pi as you will have to learn them both at the same time. The Arduino has been around for a while though, and there are tons of useful tutorials around to get you started. If you have experience in programming, you won’t be having problem in getting to grips with the Arduino straight away.

Arduino Vs Raspberry Pi

The Arduino is based on hardware, which means you won’t get far without some components: LCDs, LEDs, resistors, motors etc. Depending upon what project you want to do. You need no experience or components to get the Raspberry Pi to do something. Just plug and play.

If you want to make a hardware project, then the [Arduino is the best choice](https://www.elprocus.com/arduino-based-projects-ideas/). The input is given analog and get output is PWM and a whole spectrum of compatibility the Raspberry Pi cannot do natively. Plus the large I/O pins let you to connect the multiple sensors and feedback components. The Arduino however, is not as powerful as the Raspberry Pi, so there’s no proper video, audio, or internet out of the box. The Arduino can send data to your PC or Raspberry Pi, over serial, and you can then create a program to read this data and do something.

If you want to make a software project, then the Raspberry Pi is the way to go. The video, audio, and internet capabilities make it the winner in this aspect. There’s no need to attach external components, so there’s no real need to learn electronics.

|  |  |  |
| --- | --- | --- |
| **Feature** | **Raspberry Pi** | **Arduino** |
| Programming Language | No limit | Arduini, C/C++ |
| Processor Speed | 700 MHz, | 16MHz |
| Internet connection very easy | Not easy | Doable |
| Hardware design | Closed source | Open source |
| Real time | Hardware realtime | In real time |
| Analog to Digital | No | Yes |