Contents

[Arduino,Nodemcu and Raspberry 2](#_Toc57107959)

[Coding etiquette 2](#_Toc57107960)

[Version control 3](#_Toc57107961)

[Edge connectors 3](#_Toc57107962)

[Contacts 3](#_Toc57107963)

[Performance Specifications 3](#_Toc57107964)

[Approval Organizations 4](#_Toc57107965)

[RoHs complaint definition 4](#_Toc57107966)

[Standards 4](#_Toc57107967)

[AVX INTERCONNECT 009159010061916 Card 5](#_Toc57107968)

[Product Overview 6](#_Toc57107969)

[Applications 6](#_Toc57107970)

[AVR INTERCONNECT 009159006551906 6](#_Toc57107971)

# Arduino,Nodemcu and Raspberry

Arduino

Arduino and Nodemcu are basically development boards.It is actually made to learn how to program an MCU.We can tell the board what to do by sending a set of instructions to the microcontroller on the board.

Unlike a general purpose system such as PC a development board consist of little or no hardware for dedicates user interface.It has got only a memory to accept and run a user supplied program.

Program:C++,Embedded C.

IDE:ESPlorer,Arduino

Features:Inexpensive,simple,cheaper and opensource.

Applications:Toys,control Led brightness using PWM,Light sensor,Temperature sensor to monitor temperature of the room.

Example:remote controlled car.

Nodemcu

NodeMCU is an open source firmware for which open source prototyping board designs are available. The name "NodeMCU" combines "node" and "MCU" (micro-controller unit). The term "NodeMCU" strictly speaking refers to the firmware rather than the associated development kits.

Extra feature:Wifi connectivity

All other functions same as Arduino.

Raspberry Pi

It is a low cost ,small size computer in which a monitor,keyboard and mouse can be connected.Its capable of doing everything a normal PC can do.From browsing the internet and playing high definition video,to making spreadsheets,word processing and playing games.

Applications:Home automation,Machine learning,games,website.

Example:remote controlled car with live streaming using webcam

# Coding etiquette

1. Documentation
2. Use descriptive variable names
3. Use repeatitive codes-functions
4. Avoid multiple lines-Short handing
5. Convention(camel phrase)
6. Indenting

# Version control

1. GitHub
2. Bit bucket

Software used for version control operations is sourcetree

PCB design software used:**EAGLE**

# Edge connectors

Card edge connectors (also called edge card connectors) are embedded devices that mate with the edges of single-sided or double-sided printed circuit boards (PCBs) in order to provide an external electrical connection. Industry standards define power specifications such as voltage levels, power pin placements, and power requirements. For example, PICMG 1.0/1.2 standards define the power specifications for peripheral component interconnect (PCI) and industry standard architecture (ISA) buses.

## Contacts

The contacts are made of metal and are plated with a thin, metallic coating that improves conductivity and prevents rust and corrosion. Plating materials include:

* Gold
* Copper
* Nickel
* Silver

Contacts are housed in a termination assembly that is usually made of plastic. Card edge connectors are used widely in personal computers (PCs) and, depending on their components, are suitable for some lead-free applications and compliant with RoHS.

Rows

Typically, card edge connectors are available with one or two rows of contacts.

Contact Pitch

The distance between pins, can range from as small as 0.0197'' or 0.5 mm to as large as 0.111'' or 2.84 mm.

Geometry

There are straight and right angle card edge connectors. These distinctions simply refer to the way the card edge connector is bent.

## Performance Specifications

Includes the following; number of contacts, voltage rating, current rating, contact resistance, insulation resistance, dielectric withstanding voltage, and operating temperature.

* The voltage rating is the maximum operating voltage.
* The current rating is the maximum recommended continuous flow of electrical current.
* Contact resistance measures the electrical resistance of mated contacts that are assembled in a connector for typical service use.
* Insulation resistance is the electric resistance between two conductors separated by an insulating material.
* Dielectric withstanding voltage is the maximum potential gradient that a dielectric material can withstand without failure.
* Operating temperature is a full-required range. For specific performance specifications, contact one of the card edge connector manufacturers.

## Approval Organizations

Card edge connectors meet the requirements of a variety of national and international approval organizations. Examples include:

* Underwriters Laboratories (UL)
* The 'National Electrical Manufacturers' Association (NEMA)
* The International Electrotechnical Commission (IEC)
* TÜV Rheinland/Berlin-Brandenburg provides international approval services for product safety.
* The CSA mark indicates compliance with Canadian standards.
* CEE and VDE marks identify conformity with European guidelines.
* Japanese industrial standards (JIS)
* RoHS compliant

### RoHs complaint definition

RoHS is a product level compliance based on the European Union's Directive 2002/95/EC, the Restriction of the Use of certain Hazardous Substances in Electrical and Electronic Equipment (RoHS). Products compliant with this directive do not exceed the allowable amounts of the following restricted materials: lead, mercury, cadmium, hexavalent chromium, polybrominated biphenyls (PBB) and polybrominated diphenyl ethers (PBDE), with some limited exemptions.

This directive applies to manufacturers, authorized representatives, importers and distributors of products including large household appliances (refrigerators, etc.), small household appliances (vacuum cleaners, etc.), computing & communications equipment, consumer electronics, lighting, power tools, toys and sports equipment (videogames, electric trains, etc.) and automatic dispensers (vending machines, ATM machines, etc.).

Organizations often prove RoHS compliance with a letter of compliance issued by an employee of the company. There are third party testing services available that will test an organization's products for levels of the restricted materials.

## Standards

BS CECC 75100 - Quality assessment for electronic components: two-part and edge socket connectors for printed circuit board applications.

IEC 60603-5 - Connectors for frequencies below 3 MHz for use with printed circuit boards part 5: Edge-socket connectors and two-part connectors for double-sided printed boards with 2.54 mm (0.1 in) spacing.

# Communication Protocol

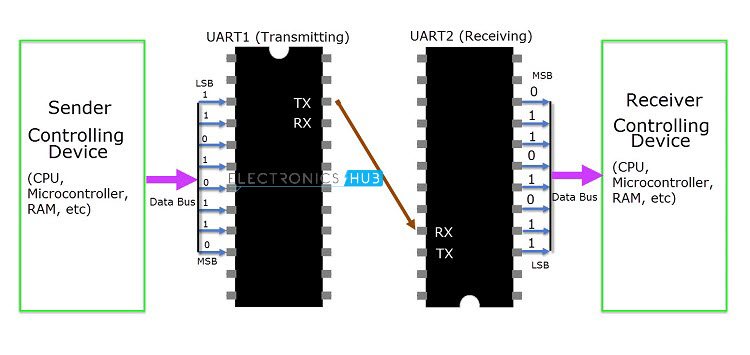
## UART-[Click here1](https://www.electronicshub.org/basics-uart-communication/)[Click here2](https://microcontrollerslab.com/uart-communication-working-applications/)

* Parallel to serial communication,Two wire protocol(Rx,Tx)
* Parallel to serial conversion in transmitter side and serial to parallel in receiver side.
* Full duplex opertion
* Less transfer speed
* No clock signal.
* Multiple master or slave is not possible

**Working:** In UART Serial Communication, the data is transmitted asynchronously i.e. there is no clock or other timing signal involved between the sender and receiver. Instead of clock signal, UART uses some special bits called Start and Stop bits.

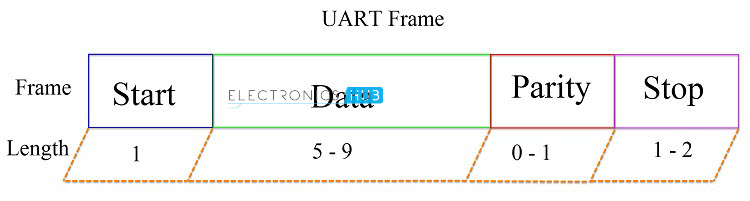
These bits are added to the actual data packet at the beginning and end respectively. These additional bits allows the receiving UART to identify the actual data.

* Step1: The receiving UART receives data from the data bus in parallel.
* Step 2: The transmitting UART adds the start, parity and stop bit to the data packet.
* Step 3: The entire packet is sent from the transmitting UART to the receiving UART serially. Using the configured baud rate the receiving UART samples the data packet.
* Step 4: The receiving UART converts the data back to its original form and then transfers it to the data bus where it can be used or visualized.



**Structure of data frame:**

The UART chip adds the start bit,stop bit and parity bit and converts serial data into parallel data using shift registers.

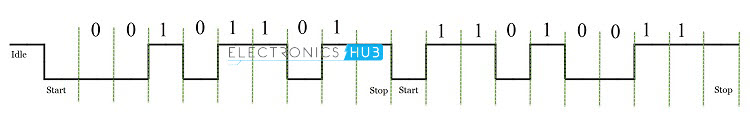


**Rules of UART**

* Synchronization bits(start bit,stop bit)
* Parity bits
* Data bits
* Baud rate: The speed at which the data is transmitted is mentioned using Baud Rate. Both the transmitting UART and Receiving UART must agree on the Baud Rate for a successful data transmission.

Baud Rate is measured in bits per second. Some of the standard baud rates are 4800 bps, 9600 bps, 19200 bps, 115200 bps etc. Out of these 9600 bps baud rate is the most commonly used one.

Let us see an example data frame where two blocks of data i.e. 00101101 and 11010011 must be transmitted. The format of the frame is 9600 8N1 i.e. 9600 bps with 8 bits of data, no parity and 1 stop bit. In this example, we haven’t used the parity bit.



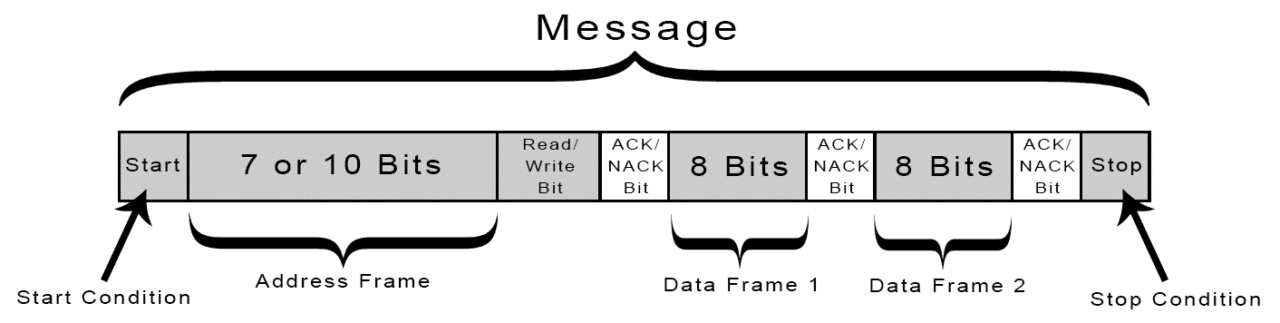
**Applications**

* Bluetooth,GPS modules.
* RFID based applications.

## I2C-[Click here](https://www.circuitbasics.com/basics-of-the-i2c-communication-protocol/)

* Two wire communication(SDA,SCL)
* Four Modes of transfer speed
* Serial communication

**Working:**Data is transferred in *messages.*Messages are broken up into *frames* of data. Each message has an address frame that contains the binary address of the slave, and one or more data frames that contain the data being transmitted. The message also includes start and stop conditions, read/write bits, and ACK/NACK bits between each data frame:



**Message frame** contains start bit, stop bit, address bits, read/write bits,acknowledge/no acknowledge bit

* **Start Condition:** The SDA line switches from a high voltage level to a low voltage level *before* the SCL line switches from high to low.
* **Stop Condition:** The SDA line switches from a low voltage level to a high voltage level *after* the SCL line switches from low to high.
* **Address Frame:** A 7 or 10 bit sequence unique to each slave that identifies the slave when the master wants to talk to it.
* **Read/Write Bit:** A single bit specifying whether the master is sending data to the slave (low voltage level) or requesting data from it (high voltage level).
* **ACK/NACK Bit:** Each frame in a message is followed by an acknowledge/no-acknowledge bit. If an address frame or data frame was successfully received, an ACK bit is returned to the sender from the receiving device.

**Addressing**

The master sends the address of the slave it wants to communicate with to every slave connected to it. Each slave then compares the address sent from the master to its own address. If the address matches, it sends a low voltage ACK bit back to the master. If the address doesn’t match, the slave does nothing and the SDA line remains high.

**READ/WRITE BIT**

The address frame includes a single bit at the end that informs the slave whether the master wants to write data to it or receive data from it. If the master wants to send data to the slave, the read/write bit is a low voltage level. If the master is requesting data from the slave, the bit is a high voltage level.

**THE DATA FRAME**

After the master detects the ACK bit from the slave, the first data frame is ready to be sent.

The data frame is always 8 bits long, and sent with the most significant bit first. Each data frame is immediately followed by an ACK/NACK bit to verify that the frame has been received successfully. The ACK bit must be received by either the master or the slave (depending on who is sending the data) before the next data frame can be sent.

After all of the data frames have been sent, the master can send a stop condition to the slave to halt the transmission. The stop condition is a voltage transition from low to high on the SDA line after a low to high transition on the SCL line, with the SCL line remaining high.

**Steps of I2C data transmission**

1. The master sends the start condition to every connected slave by switching the SDA line from a high voltage level to a low voltage level *before* switching the SCL line from high to low:
2. The master sends each slave the 7 or 10 bit address of the slave it wants to communicate with, along with the read/write bit:
3. Each slave compares the address sent from the master to its own address. If the address matches, the slave returns an ACK bit by pulling the SDA line low for one bit. If the address from the master does not match the slave’s own address, the slave leaves the SDA line high.
4. The master sends or receives the data frame:
5. After each data frame has been transferred, the receiving device returns another ACK bit to the sender to acknowledge successful receipt of the frame:
6. To stop the data transmission, the master sends a stop condition to the slave by switching SCL high before switching SDA high:

## SPI