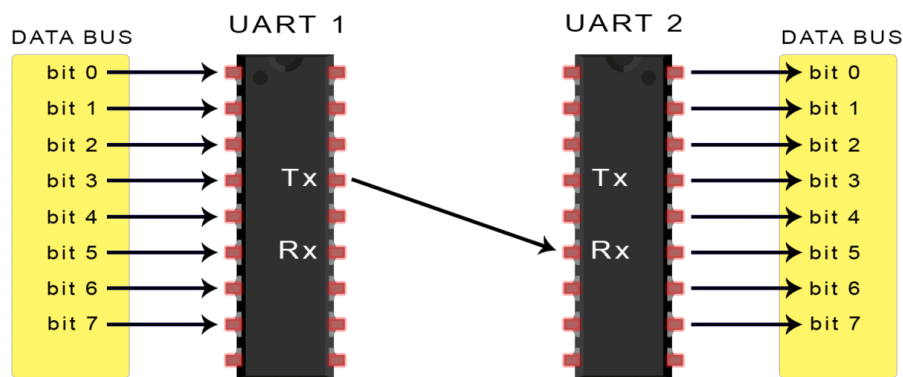


BASICS OF UART COMMUNICATION

Posted by Circuit Basics | DIY Electronics | 33



Remember when printers, mice, and modems had thick cables with those huge clunky connectors? The ones that literally had to be screwed into your computer? Those devices were probably using UARTs to communicate with your computer. While USB has almost completely replaced those old cables and connectors, UARTs are definitely not a thing of the past. You'll find UARTs being used in many DIY electronics projects to connect [GPS modules](#), [Bluetooth modules](#), and [RFID card reader modules](#) to your Raspberry Pi, Arduino, or other microcontrollers.

1 Bellek Promosyon Usb Bellek - Usb & Flash Bellek Modelleri

İstedığınız modellerde Usb & Flash Bellek Promosyon Çeşitleri Uygun Fiyatlarla bellekpromosyon.com

2 CPS Data Acquisition Software - Multiple ports data logger.

Collect data from USB/RS232 devices like bar code scanners, scales, instruments programbl.com

UART stands for Universal Asynchronous Receiver/Transmitter. It's not a communication protocol like SPI and I2C, but a physical circuit in a microcontroller, or a stand-alone IC. A UART's main purpose is to transmit and receive serial data.

One of the best things about UART is that it only uses two wires to transmit data between devices. The principles behind UART are easy to understand, but if you haven't read part one of this series, [Basics of the SPI Communication Protocol](#), that might be a good place to start.

INTRODUCTION TO UART COMMUNICATION

SEARCH ...

FOLLOW US



SUBSCRIBE

Subscribe to get new tutorials sent straight to your inbox!

EMAIL ADDRESS

SUBSCRIBE

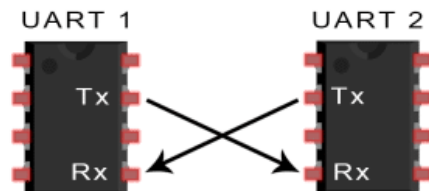
1 AM & RF Labels, and RFID Tags- - Shanghai PatronSaint Elec.Tech

Advanced EAS/RFID products cover ten fields. Free contact: yewei@patron-
eassecurity-tag.com

2 CPS Data Acquisition Software - Multi-ports data logger.

Collect data from USB/RS232 devices like bar code scanners, scales, instrum
proqrambl.com

In UART communication, two UARTs communicate directly with each other. The transmitting UART converts parallel data from a controlling device like a CPU into serial form, transmits it in serial to the receiving UART, which then converts the serial data back into parallel data for the receiving device. Only two wires are needed to transmit data between two UARTs. Data flows from the Tx pin of the transmitting UART to the Rx pin of the receiving UART:



UARTs transmit data *asynchronously*, which means there is no clock signal to synchronize the output of bits from the transmitting UART to the sampling of bits by the receiving UART. Instead of a clock signal, the transmitting UART adds start and stop bits to the data packet being transferred. These bits define the beginning and end of the data packet so the receiving UART knows when to start reading the bits.

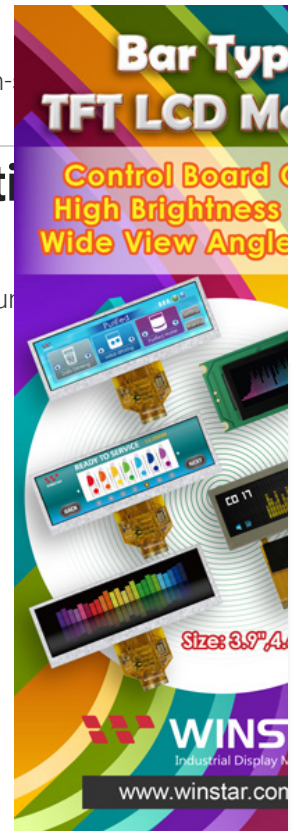
When the receiving UART detects a start bit, it starts to read the incoming bits at a specific frequency known as the *baud rate*. Baud rate is a measure of the speed of data transfer, expressed in bits per second (bps). Both UARTs must operate at about the same baud rate. The baud rate between the transmitting and receiving UARTs can only differ by about 10% before the timing of bits gets too far off.

Both UARTs must also must be configured to transmit and receive the same data packet structure.

Wires Used	2
Maximum Speed	Any speed up to 115200 baud, usually 9600 baud
Synchronous or Asynchronous?	Asynchronous
Serial or Parallel?	Serial
Max # of Masters	1
Max # of Slaves	1

HOW UART WORKS

The UART that is going to transmit data receives the data from a data bus. The data bus is used to send data to the UART by another device



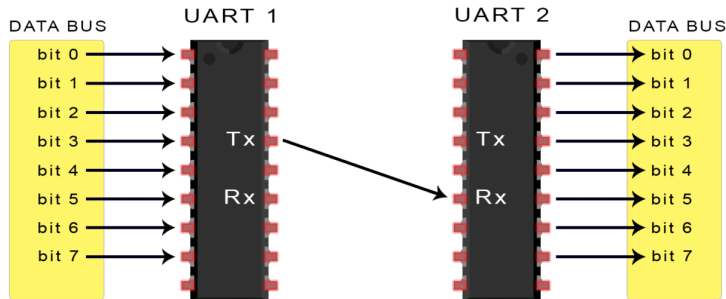
Low Cost PCB Prototype

Only **\$2** /10pcs

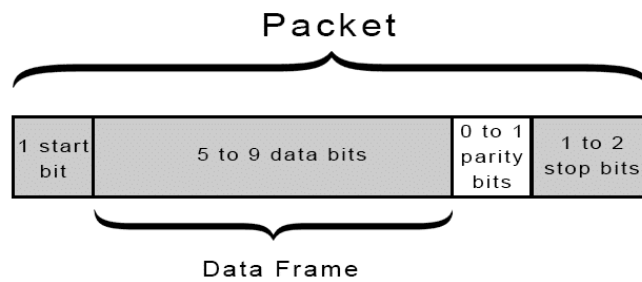
2-3 days build time >> JLPCB.COM

JLPCB is the largest PCB prototype enterprise in China with 290,000+ customers worldwide and 8000+ online orders per day.

like a CPU, memory, or microcontroller. Data is transferred from the data bus to the transmitting UART in parallel form. After the transmitting UART gets the parallel data from the data bus, it adds a start bit, a parity bit, and a stop bit, creating the data packet. Next, the data packet is output serially, bit by bit at the Tx pin. The receiving UART reads the data packet bit by bit at its Rx pin. The receiving UART then converts the data back into parallel form and removes the start bit, parity bit, and stop bits. Finally, the receiving UART transfers the data packet in parallel to the data bus on the receiving end:



UART transmitted data is organized into *packets*. Each packet contains 1 start bit, 5 to 9 data bits (depending on the UART), an optional *parity* bit, and 1 or 2 stop bits:



START BIT

The UART data transmission line is normally held at a high voltage level when it's not transmitting data. To start the transfer of data, the transmitting UART pulls the transmission line from high to low for one clock cycle. When the receiving UART detects the high to low voltage transition, it begins reading the bits in the data frame at the frequency of the baud rate.

DATA FRAME

The data frame contains the actual data being transferred. It can be 5 bits up to 8 bits long if a parity bit is used. If no parity bit is used, the data frame can be 9 bits long. In most cases, the data is sent with the least significant bit first.

1. Cheap Trucks for Sale

6. Big Data Analytic Tools

2. Electronic Circuit

7. Data Recovery Services

3. Data Entry Programs

8. Data Protection

4. Data Warehousing

9. Electronic Data

5. Data Storage

10. Dell Data Recovery



PARITY

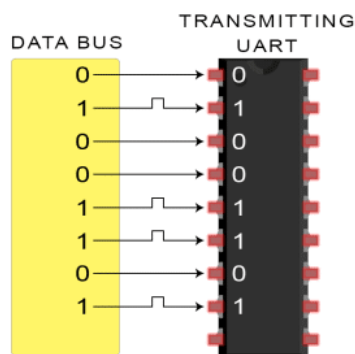
Parity describes the evenness or oddness of a number. The parity bit is a way for the receiving UART to tell if any data has changed during transmission. Bits can be changed by electromagnetic radiation, mismatched baud rates, or long distance data transfers. After the receiving UART reads the data frame, it counts the number of bits with a value of 1 and checks if the total is an even or odd number. If the parity bit is a 0 (even parity), the 1 bits in the data frame should total to an even number. If the parity bit is a 1 (odd parity), the 1 bits in the data frame should total to an odd number. When the parity bit matches the data, the UART knows that the transmission was free of errors. But if the parity bit is a 0, and the total is odd; or the parity bit is a 1, and the total is even, the UART knows that bits in the data frame have changed.

STOP BITS

To signal the end of the data packet, the sending UART drives the data transmission line from a low voltage to a high voltage for at least two bit durations.

STEPS OF UART TRANSMISSION

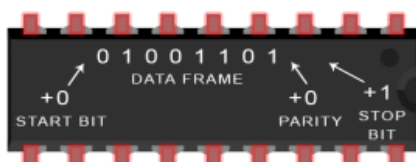
1. The transmitting UART receives data in parallel from the data bus:



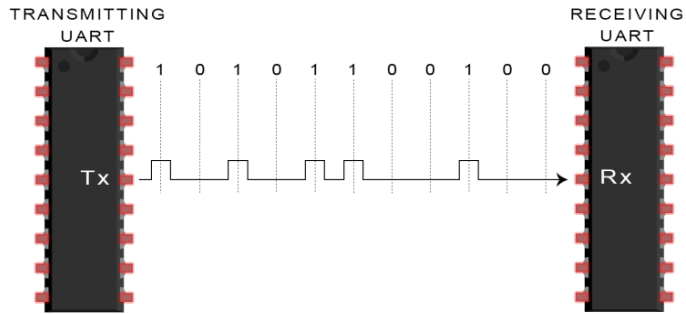
2. The transmitting UART adds the start bit, parity bit, and the stop



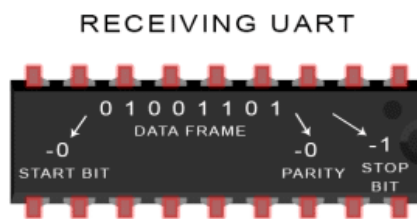
TRANSMITTING UART



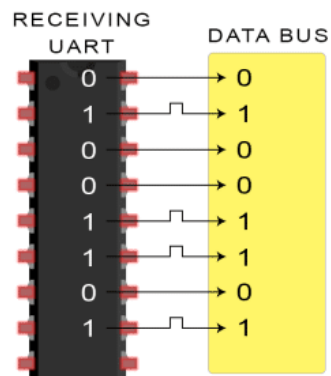
3. The entire packet is sent serially from the transmitting UART to the receiving UART. The receiving UART samples the data line at the pre-configured baud rate:



4. The receiving UART discards the start bit, parity bit, and stop bit from the data frame:



5. The receiving UART converts the serial data back into parallel and transfers it to the data bus on the receiving end:



ADVANTAGES AND DISADVANTAGES OF UARTS

No communication protocol is perfect, but UARTs are pretty good at what they do. Here are some pros and cons to help you decide whether or not they fit the needs of your project:

ADVANTAGES

- Only uses two wires
- No clock signal is necessary
- Has a parity bit to allow for error checking
- The structure of the data packet can be changed as long as both sides are set up for it
- Well documented and widely used method



DISADVANTAGES

- The size of the data frame is limited to a maximum of 9 bits
- Doesn't support multiple slave or multiple master systems
- The baud rates of each UART must be within 10% of each other

Continue on to part three of this series, [Basics of the I2C Communication Protocol](#) to learn about another way electronic devices communicate. Or if you haven't already, check out part one, [Basics of the SPI Communication Protocol](#).

And as always, let us know in the comments if you have questions or anything else to add! If you liked this article and want to see more like it, be sure to subscribe- we send out an email when ever we publish new posts.

JLPCB - Prototype PCBs for \$2 + Free Shipping on First Order

China's Largest PCB Prototype Manufacturer, 290,000+ Customers & 8000+ Online Orders Per Day

10 PCBs Price: \$2 for 2-layer, \$15 for 4-layer, \$74 for 6-layer

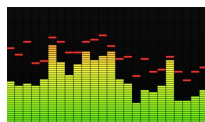
SHARE:     

ABOUT THE AUTHOR

Circuit Basics

RELATED POSTS



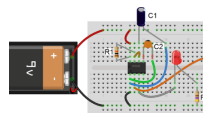
Build a Great Sounding Audio Amplifier (with Bass Boost) from the LM386



A Complete Guide to Design and Build a Hi-Fi LM3886 Amplifier



How to Design and Build an Amplifier With the TDA2050



555 Timer Basics – Monostable Mode



1. Data	6. Data
2. Cheap	7. USB Data
3. Electronic	8. Data
4. Electronic	9. Programmi
5. Dell Data	10. Data Entry



33 COMMENTS

- vivek** on June 2, 2016 at 4:43 am

Nice one. But there is a scope of introducing RS232 concept here

REPLY
- Taskeen** on September 10, 2017 at 1:54 pm

Thank you for this beautiful explanation.. Great

REPLY
- Dinesh Yadav** on September 26, 2016 at 7:09 am

Its Beautifully Explained. Frankly Speaking, After a hell lot of search over Internet and books, THis is the only Place where I literally Understood what Exactly UART is and its working.

Thanks a Lot guys.

A request: can you write another article to Explain How Exactly an IC (made of transistors) Adds these bits (Start,parity,stop) and Removes them. I mean how exactly they do this intelligent stuff at Electronics level ? Please explain its Circuit basics. 😊

REPLY
- VIKAS S DEV** on September 26, 2016 at 7:21 am

Well Explained and Nice Article.Thank You 😊

REPLY
- rekula raviteja** on October 4, 2016 at 12:36 pm

It's a good explanation.

REPLY
- Ramesh Mote** on October 8, 2016 at 8:00 pm

its very good explanation, thanks a lot



its very good explanation. thanks a lot.

REPLY

6. **Volthaus Electronics Laboratory** on October 11, 2016 at 3:50 am

Excellent.

REPLY

7. **Randall** on October 24, 2016 at 4:34 pm

This was helpful but I'm unsure about one thing. If two bits gets flipped in error, wouldn't the total still match with the parity bit? Or more generally, if an even number of bits get flipped in error, the error won't be detected. Is that right?

REPLY

Ricky on October 27, 2016 at 6:32 pm

Correct, the receiver will not notice the error, but, pending good design of the hardware, this is going to be a rare case.

REPLY

ArchonOSX on December 18, 2016 at 2:17 pm

Your analysis is correct. That is why parity checking is rudimentary and may not prevent all errors. But the odds of two bits being changed is much higher than just one so this method works in most cases of low level hardware communication.

REPLY

8. **Olivia Christy** on November 4, 2016 at 7:01 am

Nice work. As a begginer this helps me a lot .can you do this for uart with interrupts also

REPLY

9. **rajkumar** on November 30, 2016 at 9:58 am

Well explanation

REPLY

10. **hardik tank** on December 3, 2016 at 6:10 am

well explanation,and its easy to understand
friendly explanation „nice
thanx



REPLY

11.

SARS on December 7, 2016 at 4:44 am

really easy of understanding and way of explanation is good

REPLY

12.

haleem on December 11, 2016 at 3:38 pm

Nice one!

REPLY

13.

vj on December 16, 2016 at 1:05 pm

very good expaination. I have one question that we also use Usb micro port for connectiing uart hardware and the other USB port type A to our laptop. How does it work? does uart support USB ports also?

REPLY**Harshal** on January 4, 2017 at 10:40 am

no uart does not support the usb ports

REPLY

14.

OF on December 20, 2016 at 11:07 pm

"It's not a communication protocol like SPI and I2C, but a physical circuit in a microcontroller, or as a separate IC".

This part confuses me. I understand that UART is a physical hardware IC, but it must also be a protocol because how else would there be communication? The bits can't just be sent randomly. Data is sent through packets just like you described in a timely matter and specific order. Isn't that in essence a protocol? Thanks.

REPLY

15.

rk yadav on December 22, 2016 at 5:40 am

good explanation to thanks

REPLY

16.

karthik on January 2, 2017 at 11:02 am

thanks for the good information. simple and best.

REPLY

17.

nagul sharief on January 19, 2017 at 10:05 am

i'm not understanding of parity bit

REPLY**Harshal** on February 2, 2017 at 4:47 am

marshal on February 2, 2017 at 4:47 am

Parity bit is that it show that how many bit are there and if there are even number is there then it will pass bit 1 or bit 0

REPLY

18. **Syed Ibrahim** on March 8, 2017 at 4:19 am

Good post. Its in a language everyone can understand. But i also want to know how communication (data transfer) between data bus and UART occurs.

REPLY

19. **ankur singh** on March 21, 2017 at 10:12 am

It is really a very good explanation as i m not from electrical background.

REPLY

20. **Chandru** on March 23, 2017 at 8:33 am

Can anyone say how can I transmit 1000 bytes of data through uart protocol

REPLY

21. **k t r** on April 4, 2017 at 12:26 pm

Awsome explanation really

REPLY

22. **Hemavathi** on May 15, 2017 at 8:25 am

Really good explanation. Now i understood very clearly what exactly The UART performs internally

REPLY

23. **thoshif khan** on July 12, 2017 at 6:30 am

i am working on UART so any body can send VHDL CODE to my mail thoshifkhan1992@gmail.com

REPLY

24. **k.sudheer kumar** on July 13, 2017 at 5:11 am

It is an nice explanation .Thank you

REPLY



25. **Flash9325** on September 26, 2017 at 11:21 pm

1) The two UARTs communicating having the same baud rate, does that mean the clocks of both the UARTs are synchronized (i.e clock cycle

starts and stops at the same time) ?

I didnt get from where that allowed difference of 10% came from. Can you please explain!!!

REPLY

dinesh aitha on October 30, 2017 at 3:17 pm

10% come from start and stop bits and parity too

REPLY

RK on November 27, 2017 at 9:50 am

Thank you soo much.. Thanks a ton. Really U ppl had done a great job. Now seriously I understood the working principle, data transmission of bits and everything without any doubt. Thanks again and again. Hope you upload more and more sessions and concepts like this.

REPLY

26. **Hari** on October 30, 2017 at 6:55 am

Why we using 9bits institute data transmission

REPLY

LEAVE A REPLY

Your email address will not be published. Required fields are marked *

COMMENT

//

NAME *

EMAIL *

WEBSITE

Send new posts to my inbox.

Notify me of follow-up comments by email.

POST COMMENT

^

Copyright **Circuit Basics**

[Raspberry Pi](#) [Arduino](#) [DIY Electronics](#) [Programming](#) [Videos](#) [Resources](#) [About](#) [Contact Us](#) [Privacy Policy](#)



