Kuwait University
College of Engineering and Petroleum

جامعة الكويت
KUWAIT UNIVERSITY

ME319 MECHATRONICS

Part I: The Brains – Microcontrollers, Software and Digital Logic

Lecture 8: Communication

Spring 2021

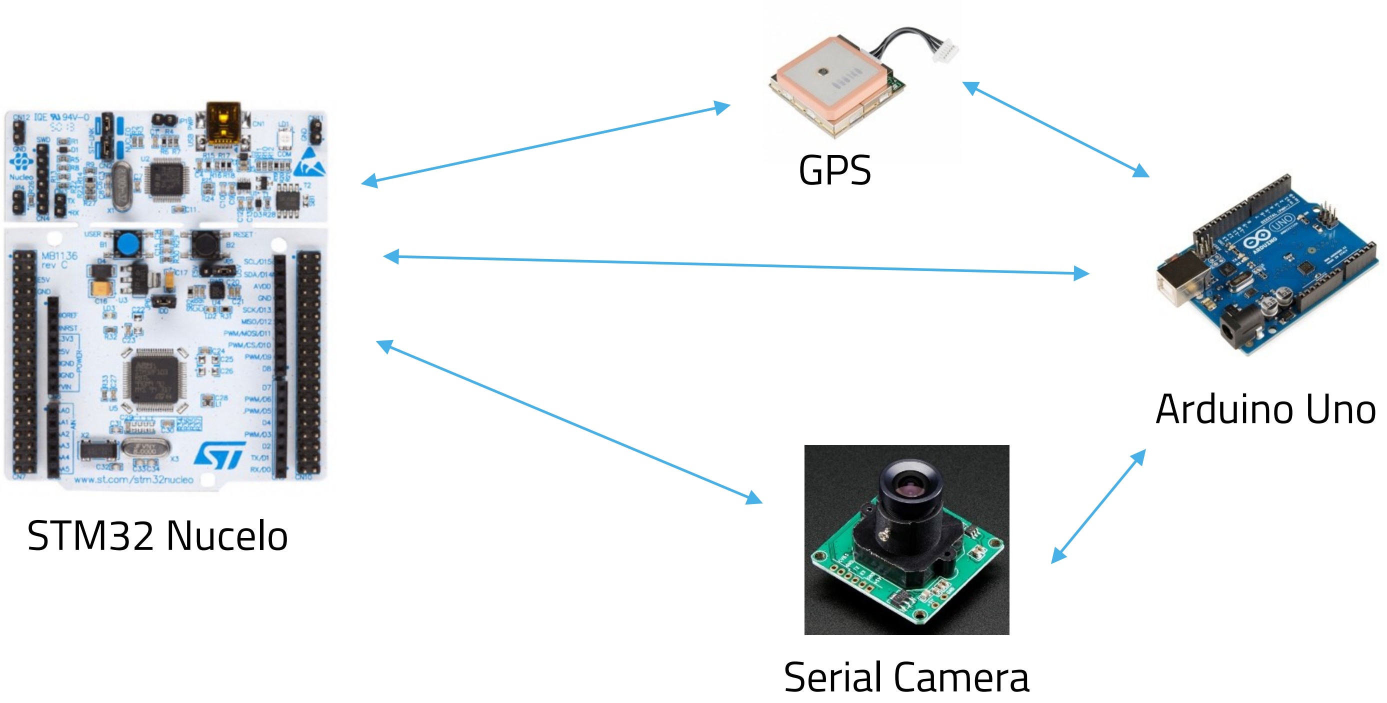
Ali AlSaibie

Objectives

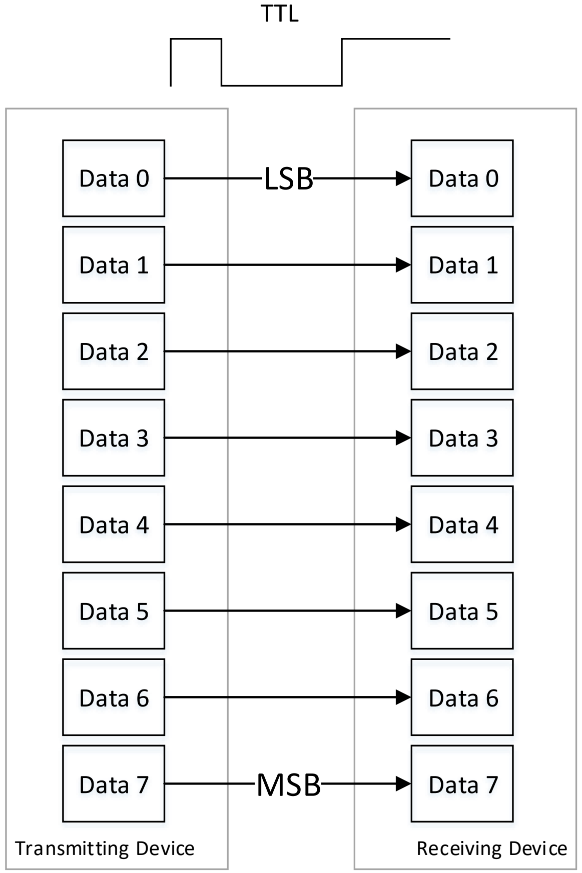
- Understand the basic form of inter-device communication
- Understand how asynchronous serial communication works

Device to Device Communication

- Some devices and sensors convey data beyond simple analog data or a few digital I/Os
  - e.g. GPS, Camera, Motor Controller, MCU to MCU

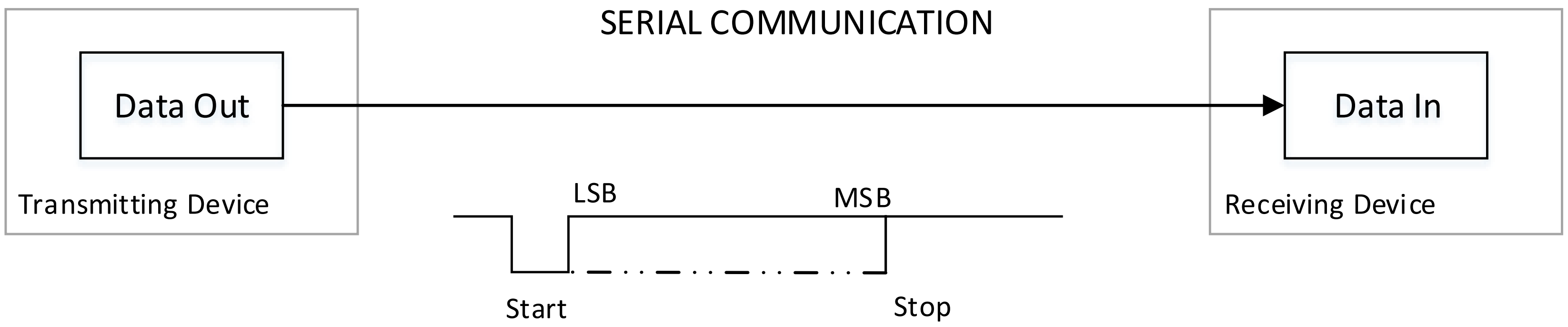


- Parallel: A group of bits are transferred concurrently
  - Bus communication on chip
  - Older printers, laptop docking stations
  - Usually 8 bits or more of data (1 word)
  - Pros: faster transfer of data (for similar frequency)
  - Cons: limited distance, high SNR, cross-talk, limited frequency



PARALLEL COMMUNICATION

- Serial: Sequential Transfer of Bits
  - Single Data in, Single Data out physical lines
  - E.g. USB, Firewire, PCI Express, RS-232, Ethernet, I2C, SPI, UART and many others
  - Pros: Less SNR and cross talk, higher frequency capable, less cost, longer range
  - Cons: Slower for lower frequencies (data rates)

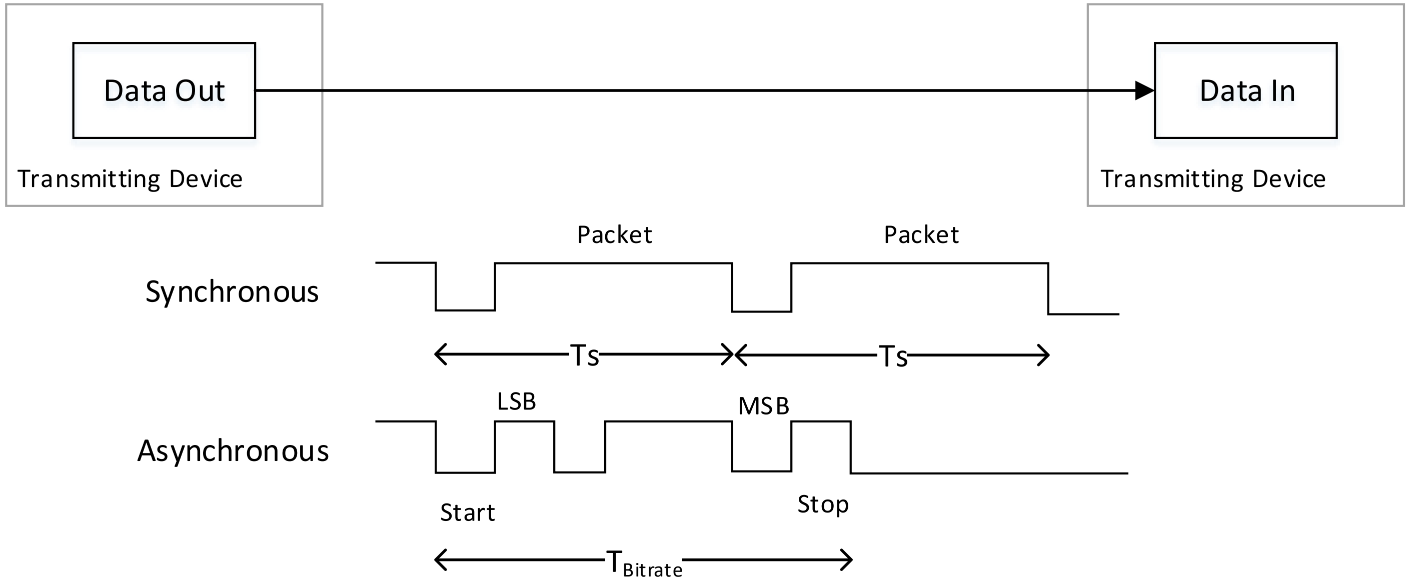


جامعة الكويت
KUWAIT UNIVERSITY

Synchronous vs Asynchronous

- Synchronous: Data transfer at a set frequency
  - Communicating devices must “synchronize” transfer frequency (timing of packets)
  - Transfer occurs regardless if new data is present
- Asynchronous: Data transfer on request
  - Devices only agree on data transfer rate (bits/s)
  - A start and stop bit must be used

- Synchronous
  + Less overhead, pure data, faster
  - Complex



- Asynchronous
  + Simple, Faster to setup
  - Larger overhead (more bits than actual data)

- Modern microcontrollers support the following most common serial communication protocols
  - UART: Universal Asynchronous Receiver/Transmitter
  - USART: Universal Synchronous-Asynchronous Receiver/Transmitter
  - SPI: Serial Peripheral Interface
  - I2C: Inter-Integrated Circuit (I-squared-C)
  - CAN: Controller Area Network
  - USB
  - Ethernet
- They differ in complexity (in hardware and/or software), range, maximum data rate, maximum channels,

جامعة الكويت
KUWAIT UNIVERSITY

Communication Speed

- Quoted in number of bit multiples per second
  - kbit/s or kb/s or kbps (kilobits per second) = 1000 bits per second
- Or binary multiples of bits per second
  - **K**ibit/s (kibibit per second)= 1024 bits per second
- Or decimal multiples of bytes per second
  - kB/s (kilobyte per second) = 8,000 bits/s = 1,000 bytes/s
- Or binary multiple of bytes per second
  - **K**iB/s (kibibyte per second) = 1024 bytes/s

Multiples of bytes v·t·e

Decimal			Binary				
Value	Metric		Value	IEC		JEDEC	
1000	kB	kilobyte	1024	KiB	**kibibyte**	KB	kilobyte
1000^2	MB	megabyte	1024^2	MiB	mebibyte	MB	megabyte
1000^3	GB	gigabyte	1024^3	GiB	gibibyte	GB	gigabyte
1000^4	TB	terabyte	1024^4	TiB	tebibyte	–	
1000^5	PB	petabyte	1024^5	PiB	pebibyte	–	
1000^6	EB	exabyte	1024^6	EiB	exbibyte	–	
1000^7	ZB	zettabyte	1024^7	ZiB	zebibyte	–	
1000^8	YB	yottabyte	1024^8	YiB	yobibyte	–	

Orders of magnitude of data

Communication Speed

- In the context of Serial communication: Baud Rate is used
- Baud Rate: Number of symbols transmitted in one second
  - Baud Rate generally means Symbols per second (not necessarily bits)
  - In serial communication, Baud Rate = Bits/s (Symbol = Bit)
- Baud rates normally take the following values:
  - 300, 600, 1200, 2400, …. 256000 bits per second
- Two devices connected through a UART channel must have the same configuration
  - Same baudrate
  - Same data bit size
  - Same parity control

جامعة الكويت
KUWAIT UNIVERSITY

UART

- A very common form of serial communication is UART
- UART: **Universal Asynchronous Receiver/Transmitter**
- Data Frame

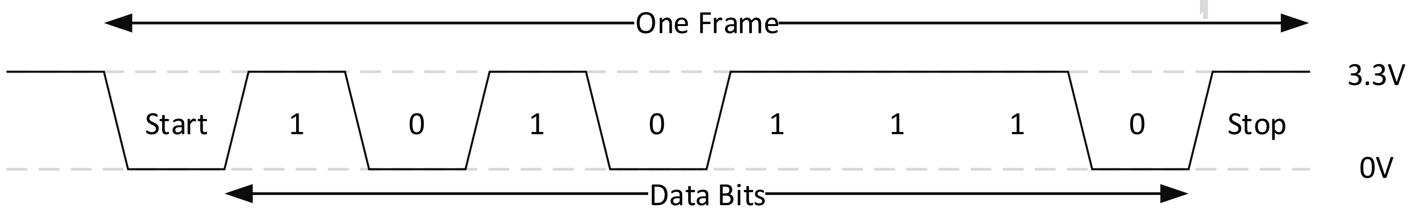
Bit #	1	2	3	4	5	6	7	8	9	10	11	12	13
	Start Bit	LSB				5-9 data bits				MSB	0-1 Parity Bit	1-2 Stop Bits	

- 8N1: Denotes 8 data bits, No Parity, 1 Stop Bit
  - 10 Bits per frame (overhead: 2 bits, meaning 8 bits of pure data)
- 8N1 is the most common UART configuration

Bit #	1	2	3	4	5	6	7	8	9	10
	Start Bit	LSB				8 data bits			MSB	Stop Bit

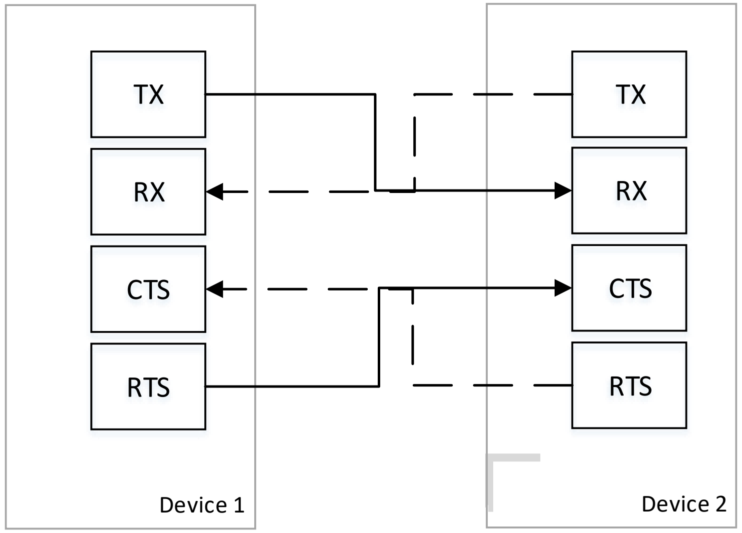
جامعة الكويت
KUWAIT UNIVERSITY

- Parity bit is used to check for errors
  - A form of CRC: Cyclic Redundancy Check
- If set, can be chosen to have odd or even parity
- Even parity: sum of 1s in data + parity bit must be even
- Odd Parity: sum of 1s in data + parity bit must be odd
- Example with Even Parity:
  - Send 0x4F (01001111): Some of 1's = 5, parity bit must be 1.
  - Receiver must check if parity bit is 1, if not there is a transfer error
- It's an optional feature



UART – Flow Control

- Devices communication on UART can coordinate data transfer through flow control
- Hardware Flow Control: Dedicated physical lines to assert availability to receive data.
- Receiver through RTS (Request-to-send) informs the other device that it's CTS (Clear-to-Send)
    - If device 1 is busy RTS will remain HIGH
    - If device 1 is free RTS will be set to LOW



UART Physical Layer

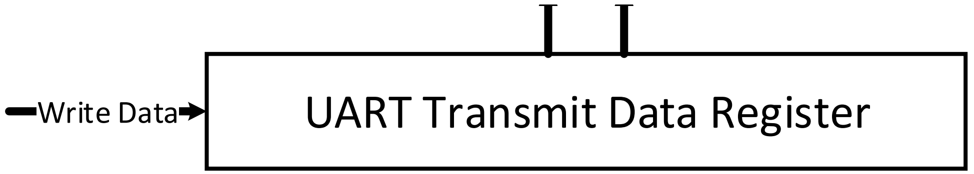
- Software Flow Control:
  - Can be attained by using interrupts
    - E.g. the receiver is programmed to read incoming data at a specific rate.
    - Or, the transmitter is programmed to only send data at specific intervals.
  - Or through special characters through data bits
    - XOFF/XON (Pause / Resume Transmission)
      - XOFF/XON has 0x13/0x11 representation
    - Software level implementation (by programmer)

جامعة الكويت
KUWAIT UNIVERSITY

- STM32F401xe includes up to 3 hardware USART peripherals
  - UART can be emulated via software using regular GPIO
    - Limited speed, software overhead
- USART: Universal **synchronous** asynchronous receiver transmitter
  - Can also support synchronous communication
- Supports 8-9 data bits
- Supports different parity settings
- 0.5, 1, 1.5 or 2 stop bits
  - 1 stop bit is the default (most common)
- Some microcontrollers have more than one buffer registers
  - Allows for queuing transmitted or received data if mcu is overloaded.
- Supports speeds up to 10.5 Mbps

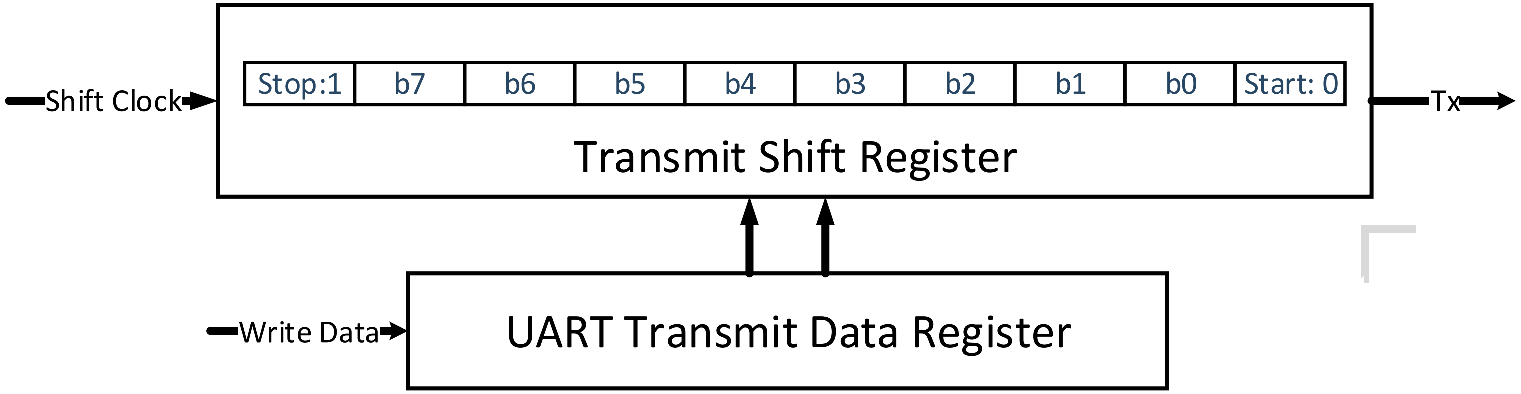
جامعة الكويت
KUWAIT UNIVERSITY

- To Transfer data through UART, data is written to UART transmit data register one byte at a time (for 8N1 configuration)



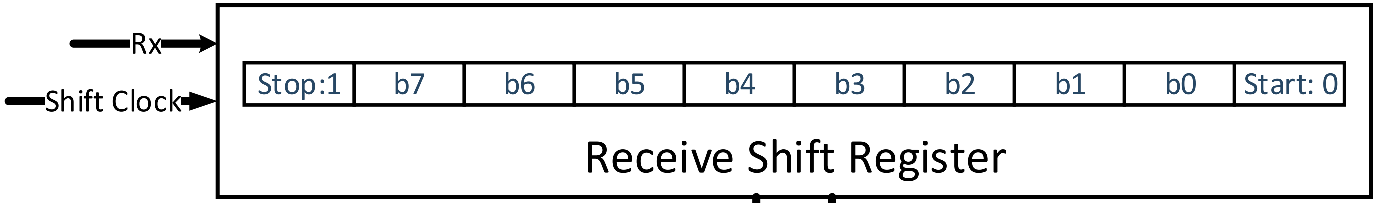
UART Transmit Process

- To Transfer data through UART, data is written to UART transmit data register one byte at a time (for 8N1 configuration)
- Data is transferred into the transmit shift register
- When the transmit bit (TE) is enabled, data on the shift register is transmitted out on the TX pin
  - First out: start bit, then LSB, etc, and finally the stop bit



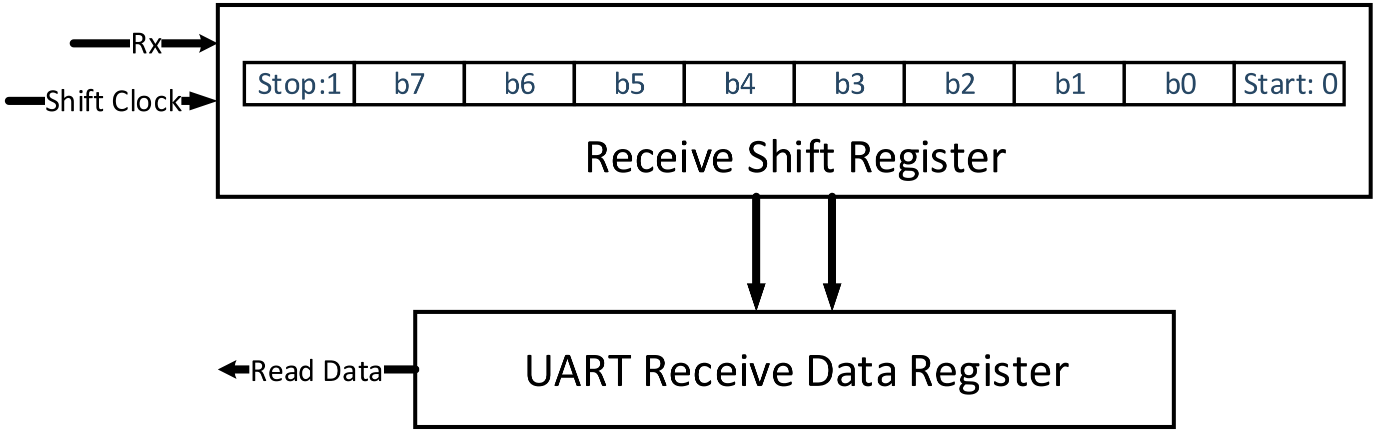
UART Receive Process

- The UART recognizes a start bit and stop bit as they come into the shift register
- The bits are shifted in the same order as the transmitter, Start,LSB .. MSB, Stop



UART Receive Process

- The UART recognizes a start bit and stop bit as they come into the shift register
- The bits are shifted in the same order as the transmitter, Start,LSB .. MSB, Stop
- Data is transferred from the shift register to the receive data register
  - A flag bit (RXNE) is set: indicating the availability of new data.
  - An interrupt can be generated (what's the benefit?)



Communication Speed - Example

- Given a 640x480 pixel 8-bit grayscale uncompressed image
- With a Baud Rate of 9600 and 8N1 UART
- How long would it take to transfer an image?
  - Assuming an un-interrupted transfer

  Image size in bytes : 640x480 = 307,200 bytes

  Data Rate: $\frac{9600\ bits/s}{10 bits/databyte} = 960\ databyte/s$

  Image transfer rate = $\frac{307{,}200}{960}$ =320 seconds/image

  With 256000 Baud Rate -> =$\frac{307{,}200}{25600} = 12\ seconds/image$

- Hence the importance of image compression
  - JPEG compression can shrink the image data size down to 2% of uncompressed size
- That's why it's hard to transfer and process images in real time on a microcontroller
  - Limited compression/decompression ability and slow data transfer and memory capacity for raw images.

جامعة الكويت
KUWAIT UNIVERSITY

Example 1

What is the minimum baud rate required to transmit the following array at 2000Hz?

If 8N1 UART frame is used.

```
Uint32_t array[30];
```

جامعة الكويت
KUWAIT UNIVERSITY

```
#include <Arduino.h>
void setup(){
    Serial.begin(250000); /* 250000 is the baud rate */
    /* On Reading Characters wait for a max of 10ms for new characters */
    Serial.setTimeout(10);
    /* Flush the UART buffer, good for clearing backlog of characeters and only
     * caring about the latest received values */
    Serial.flush();

    /* Read from the buffer until a new line '\n' is detected */
    char buffer[10];
    Serial.readBytesUntil('\n', buffer, 10);

    /* Change which pins are TX/RX (must be compatible) */
    Serial.setTx(PB6);
    Serial.setRx(PB7);

    Serial.getTimeout(); /* */

    Serial.print("Hi"); /* Print without return line */
    Serial.println("Hi"); /* Print with return line */

    /* print with printf formatting */
    char buffer2[20];
    sprintf (buffer, "It's %d in the morning", 10.0);
    Serial.print(buffer);
};
void loop(){};
```