EG3576

COMMUNICATIONS ENGINEERING I
- COMMUNICATIONS FOR CONTROL

GORRY FAIRHURST
SCHOOL OF ENGINEERING
UNIVERSITY OF ABERDEEN

HTTP://WWW.ERG.ABDN.AC.UK/~GORRY/EG3576/

V78 (JAN 2024)

AIMS

According to the web site, the aims are:

- To present the fundamentals of **serial communications** and use to control real-world equipment.
- To gain an understanding of serial data techniques using **asynchronous and synchronous** transmission and related software algorithms.
- To become familiar with the operation of **key protocols** (e.g. DMX-512, RDM, CAN).
- To introduce a professional oscillosope and use this to **measure the signal** at the bus interfaces.

The course roadmap is on-line at:

https://erg.abdn.ac.uk/users/gorry/eg3576/

INTRODUCTION TO THE COURSE

Module 0.0

KEY TO SLIDES

- YELLOW SLIDES ARE ONLY FOR LECTURER USE

Expect lectures/tutorials/ in all timetabled slots !!!

20 Lectures + 10 Tutorials + 8 Labs*

** Attendance at labs is compulsory and non-attendance will prevent submission of a continuous assessment mark.*

Communications Engineering I: Modules

0.0 Overview
- 0.1 Scopes
- 0.2 Long Distance Communications

1.0 Asynchronous Serial Transmission
- 1.1 Asynchronous Transmission
- 1.2 UART
- 1.3 EIA-232

2.0 Communications Links
- 2.1 Asynchronous Serial Frames
- 2.2 NMEA GPS Frames
- 2.3 Transmission Theory

3.0 EIA-485 Differential Transmission
- 3.1 Differential Transmission
- 3.2 EIA-485 Cable Bus

4.0 DMX 512 Physical Layer
- 4.1 DMX 512 Overview
- 4.2 Bus Terminatiion
- 4.3 Bus Transceivers

5.0 DMX 512 Frames
- 5.1 Frames of Slots
- 5.2 Addressing and Receivers
- 5.3 DMX Receiver Hardware
- 5.4 DMX Receiver Software
- 5.3 Digital Control

6.0 DMX 512 Control
- 6.1 Controlling Power
- 6.2 System Architecture
- 6.3 Multiple Slots
- 6.4 LEDs
- 6.5 Start Codes

7.0 Control Networks
- 7.1 Repeaters
- 7.2 Ethernet

8.0 RDM

9.0 CAN
- 9.1 CAN Physical Layer
- 9.2 CAN Arbitration

Communications Engineering I: Tutorials

Tutorials Topics

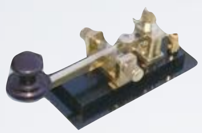
- *Asynchronous Transmission and Reception*
- *UARTs*
- *DMX Slot Transmission*
- *DMX Frame Transmission*
- *DMX Microcontroller Algorithms*
- *Remote Device Management (RDM)*
- *Controller Area Network (CAN) Bus*

LONG DISTANCE SERIAL COMMS

Module 0.1

SENDING WORDS USING ELECTRICITY

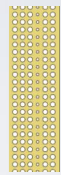
1837



... - - - .

1844

1865



1874

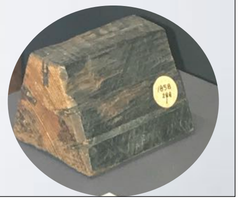
TELEGRAPHY (1837)

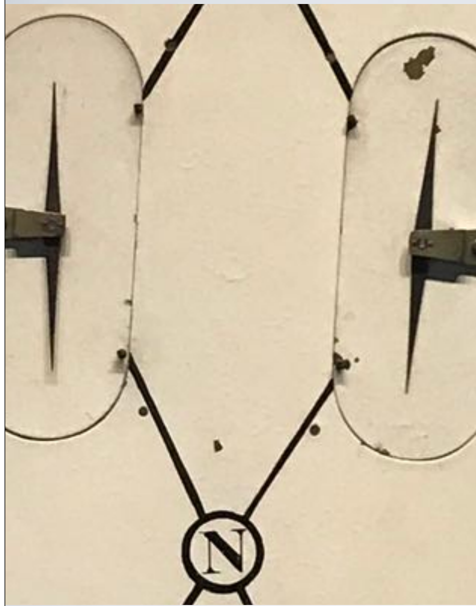
COOKE AND WHEATSTONE TELEGRAPH (PARALLEL SYMBOLS)

5 needle code

Pairs of needles pointed at a letter

6 parallel wires





MORSE CODE (1844)

SERIAL COMMUNICATIONS (TEMPORAL CODE)

3 symbols, dot, dash, space

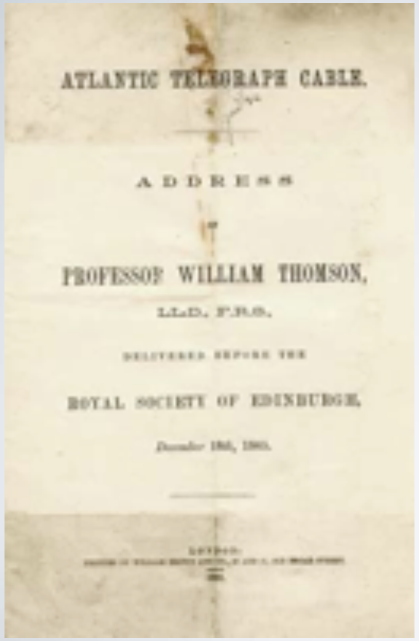
One wire or radio channel

Letters and numbers are serialised into a single stream

ATLANTIC TELEGRAPH (1857,1865)

ATLANTIC TELEGRAPH (1857,1865)

ATLANTIC TELEGRAPH (1857,1865)


ATLANTIC TELEGRAPH CABLE.

ADDRESS

PROFESSOR WILLIAM THOMSON,

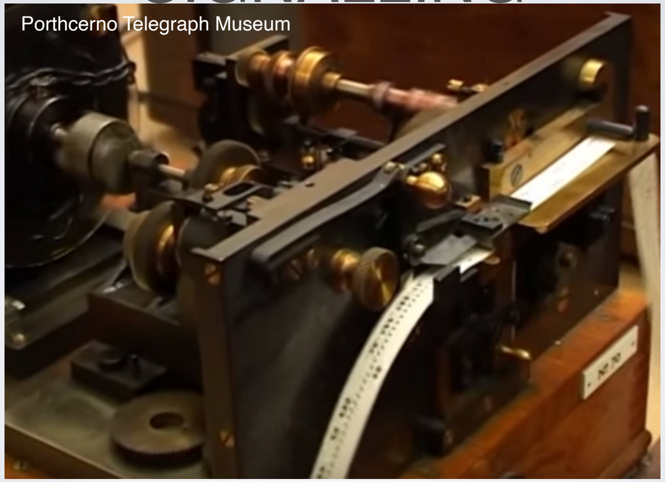
LL.D. F.R.S.

DELIVERED BEFORE THE

ROYAL SOCIETY OF EDINBURGH,

1857 1858 1865

BIPOLAR MORSE SIGNALLING



BAUDOT: 5-BIT CODE

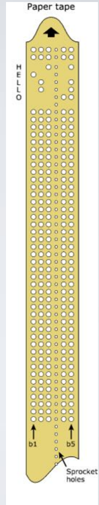
2^5 = 32 values

26 Letters

4 Control Chars
- Null (0)
- Space
- Carriage Return
- Line Feed

2 Shift Chars
- Number Shift*
- Letters Shift*
  - *26 Numbers
  - *26x2

Telex paper tape

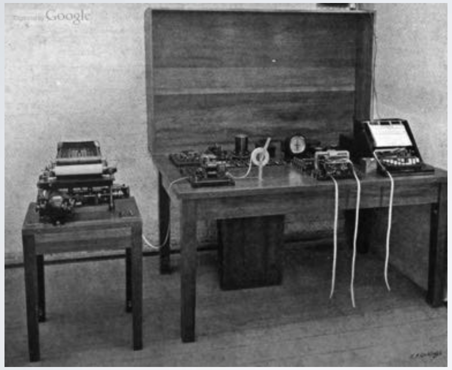


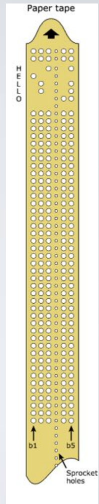
5-BIT BAUDOT CODE (1874)

00000	00000	Null
00100	00100	Space
10111	11101	Q
10011	11001	W
00001	10000	E
01010	01010	R
10000	00001	T

All characters represented by 5-bit values
5-bits represent (2^31)-1 different characters = 31.

MURRAY PRINTING TELEGRAPH (1905)





A start-stop system using 5-bit codes

MURRAY TELEPRINTER

A start-stop system using 5-bit codes

TTL ← 0 1 0 1 0 0 1 1

Start: 0

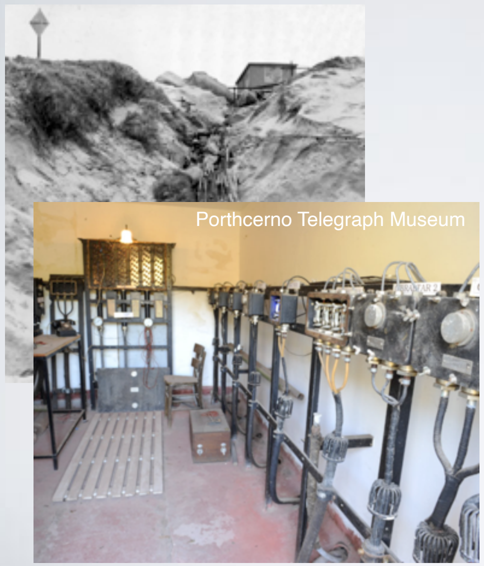
Data: 00101 ('H' in ITA2)

Stop: 11

CREED MK3 TELEPRINTER

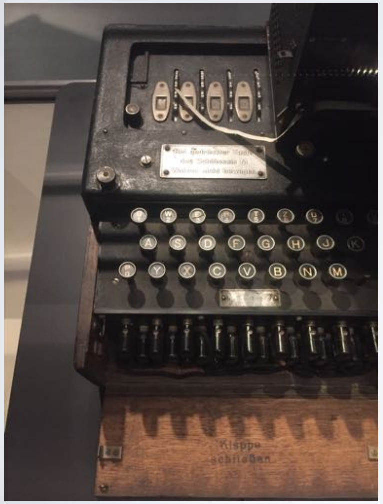
USED THROUGHOUT THE UK FOR SENDING/RECEIVING TELEGRAMS

PORTHCERNO, CORNWALL



ENIGMA (1928)

ELECTROMECHANICAL ENCODER



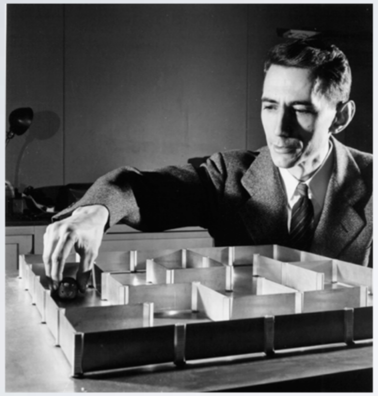
BRITISH TYPEX (1937)

ELECTROMECHANICAL TELEX MACHINE

ALAN TURING (1940'S)

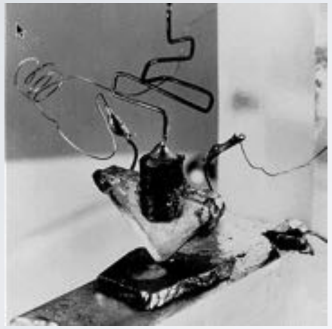
CLAUDE SHANNON (1948)

INFORMATION THEORY



Introduced the term "bit"

FIRST TRANSISTOR (1947)

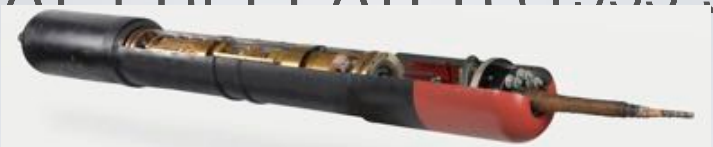


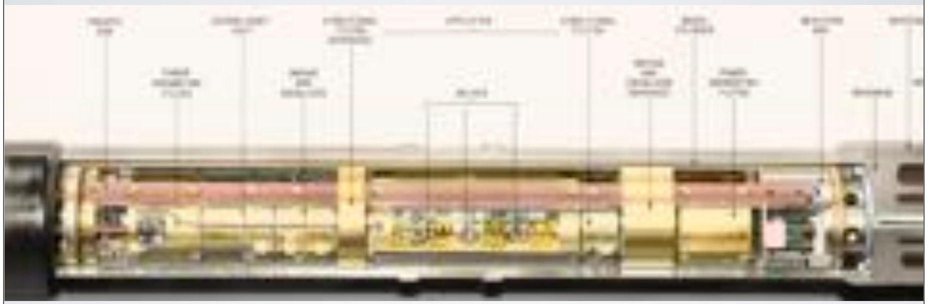
John Bardeen, William Shockley and Walter Brattain

GALLANACH BAY,
SOUND OF KERRERA NEAR OBAN

GALLANACH BAY,
SOUND OF KERRERA NEAR OBAN

TAT-1 REPEATER (1955-56)



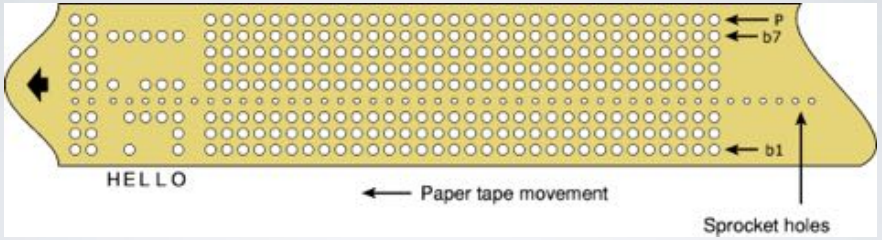


Using low-loss coaxial cable, signal could be sent 69 km

- Photos: Science Museum Lond

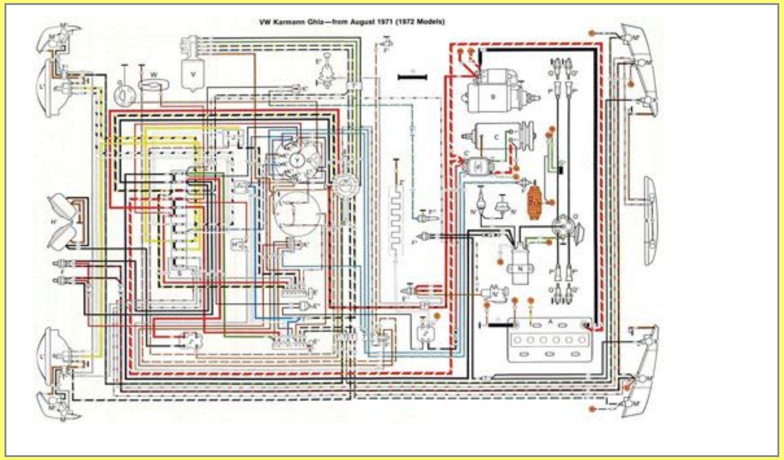
ASCII (1963)

AMERICAN STANDARD CODE FOR INFORMATION

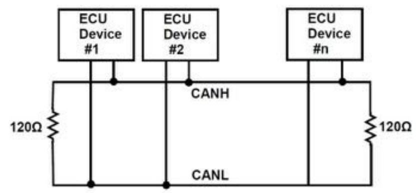


An 7-bit code (1 bit “spare”)

WIRING DIAGRAM VW BEETLE (1972)



MULTIPOINT CONTROL: CAN BUS (1986)



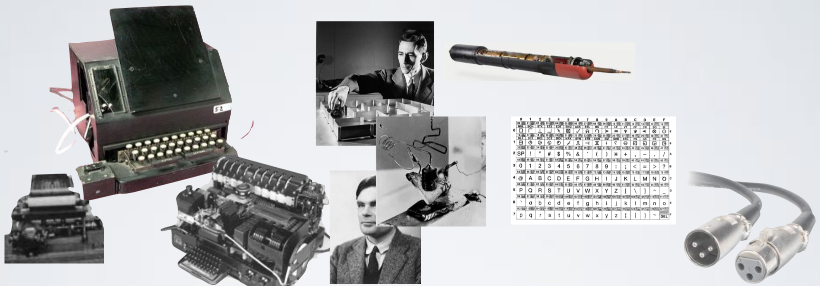
Before CAN …
there could be miles of cable in a car

MICROCONTROLLERS &

CAN IN SPACEFLIGHT

ECSS-E-ST-50-15C (May 1, 2015

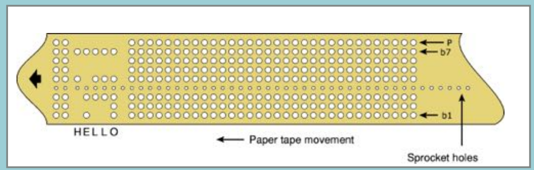
AUTOMATING COMMUNICATION 1900'S



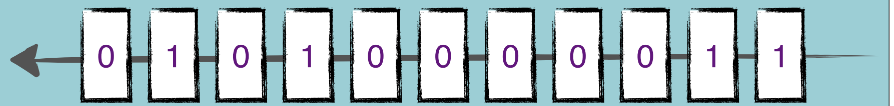
1905 1931 1942 1948 1956 1963 1986 1990s

ASYNCHRONOUS SERIAL TRANSMISSION

Module 1.0

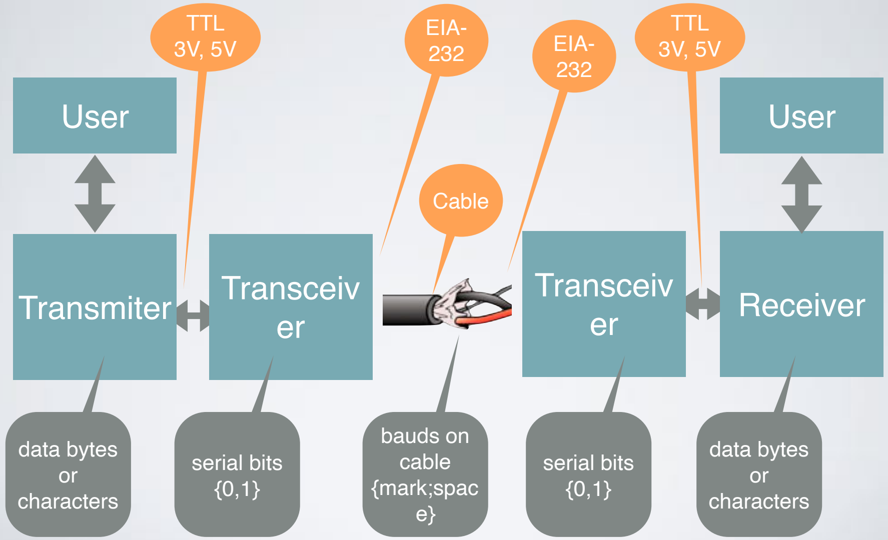


CHARACTER TRANSMISSION



Module 1.1

TRANSMISSION SYSTEM



BAUDS AND BITS

BAUD* - Number of physical transitions per second on a cable

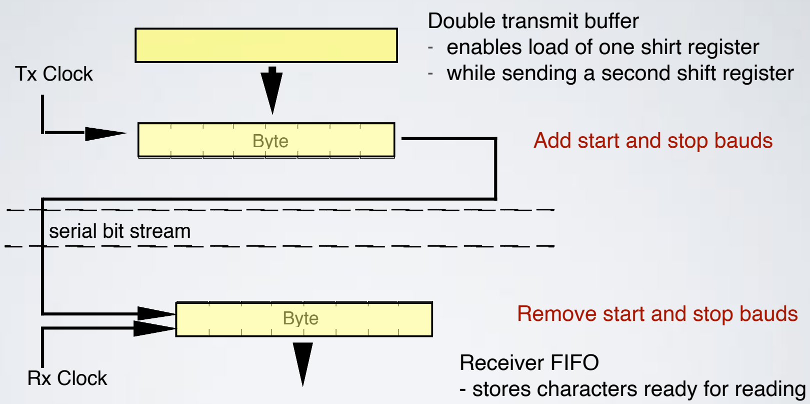
If one bit is sent in each baud, then the ***baud rate*** and ***bit rate*** would be the same.

This is **not** the case for asynchronous transmission!

8 bits are sent in 11 bauds.

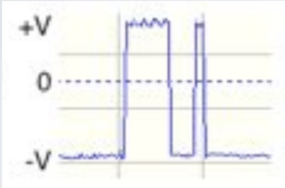
** named after JME Baudot*

SERIAL COMMUNICATIONS



Uses two shift registers (both clocks must be the same)
- Note that bytes are sent l.s.b. first!

EIA-232 SIGNAL LEVELS SENT ON THE CABLE



EIA-232

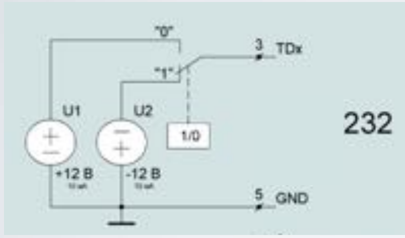
O baud - negative voltage (-12V)
1 baud - positive voltage (+12V)
Both voltages referenced to GND

TTL

Digital Interface
0 - baud below threshold
1 - baud above threshold

The line driver ***inverts*** the signal and ***changes*** the voltage

MODEL FOR EIA-232



EIA 232 switches between a positive and negative voltage depending on the baud value

ASCYNCHRONOUS SLOT (CHAR) FRAMING

Data is organised in bytes/slots and then serialised to bits

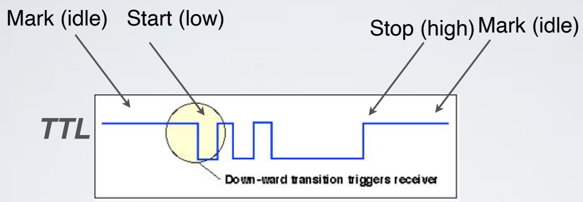
Sender and receiver both know the rate of transmission

- Each has a digital clock set to the same nominal baud rate
- This clock determines the duration of each baud
- The clock signal is NOT sent to the receiver

How can the receiver know when each byte starts?

ASYNCHRONOUS SLOTS

Data set in a slot. Let's look at how one slot is sent...



Arbitrary idle gap between slots, uses Mark level (*high*)

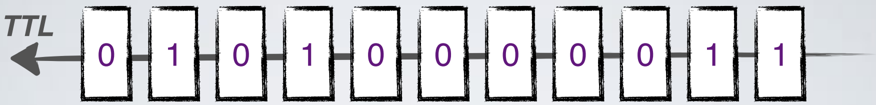
Each slot starts with one start baud (*low*)

The bits in a byte/slot are sent LSB first (bit order reversal)

Each slot ends with two stop bauds (*high*)

Shows only "A" signal

ASYNCHRONOUS SLOTS



$f = \frac{1}{T}$ and $T = \frac{1}{f}$

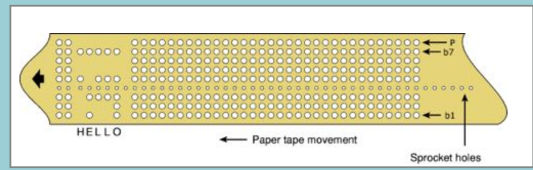
Example 1: Low speed (e.g. 4800 baud) clock

0.21 mS pulses (1/4.8 kbaud) with 8 data bits/frame

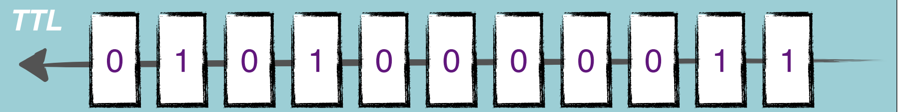
Example 2: DMX-512 sender and receiver use a 250 k baud clock

4μS pulses (1/250 kbaud) with 8 data bits/slot

Total slot duration is therefore 44 μS



ASYNCHRONOUS CHARACTER RECEPTION



Module 1.1

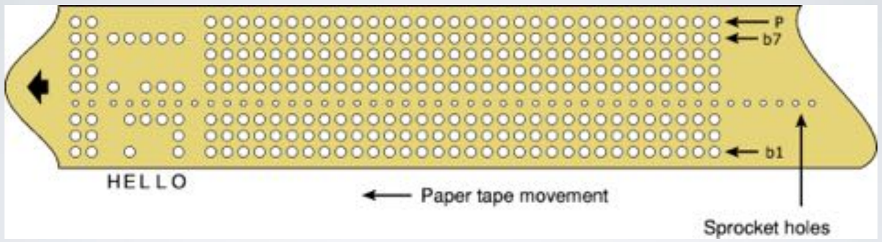
SERIAL BYTE STREAM

Multiple bytes are sent as a series of successive slots:

7654321	7654321	7654321	7654321

Text can be sent by encoding each character as a byte

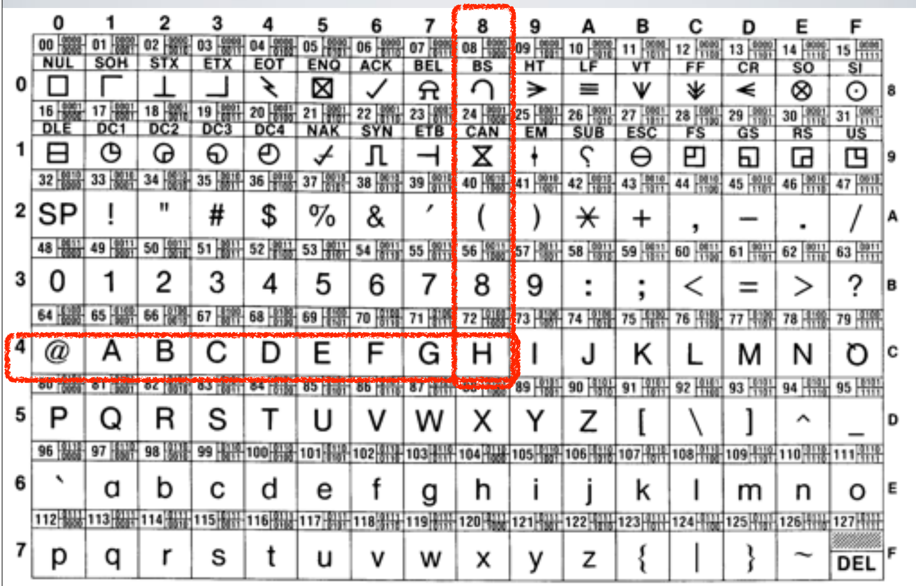
8 BIT TRANSMISSION: ASCII (1963)



An 7-bit code (1 bit "spare")

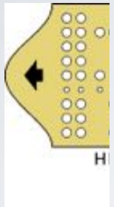
ASCIII

0x48, 72 in decimal= 'H'



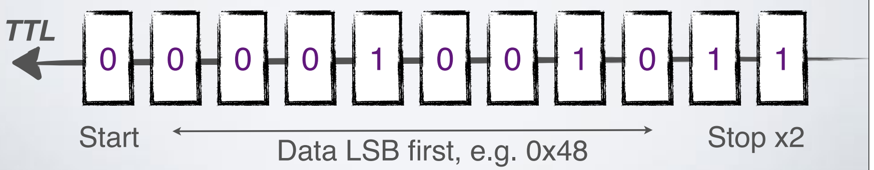
8-BIT ASCII

Punched paper tape



Data MSB first, 0100 1000 = 0x48

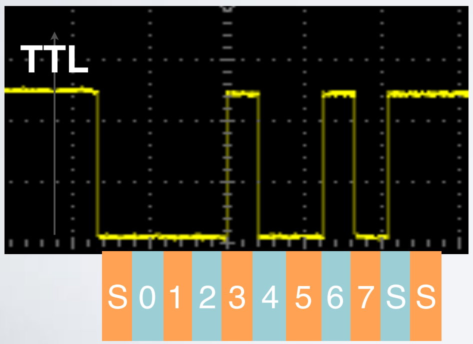
Serial Transmission



SERIAL TRANSMISSION

ASCII Character ‘H’ is 0x48 %0100 1000

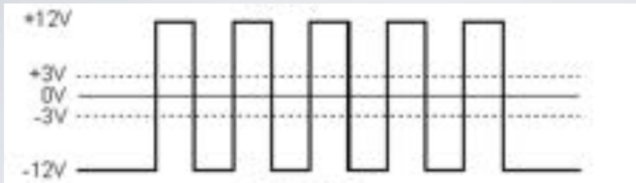
Transmission order: %(0) 0001 0010 (11)



Baud rate 9600bps

1/9600 -> 0.1 ms per baud

SERIAL RECEPTION (232)



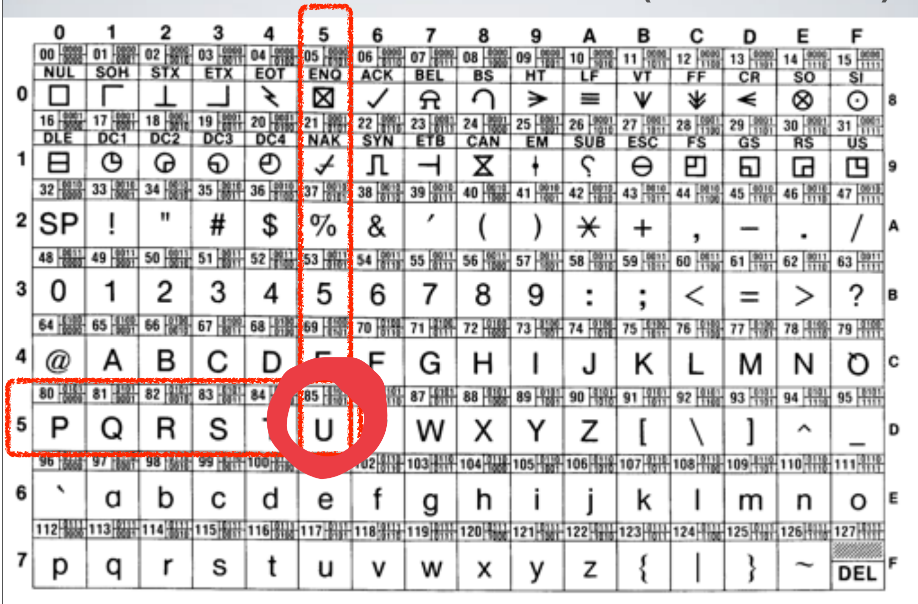
Line signal

Digital output

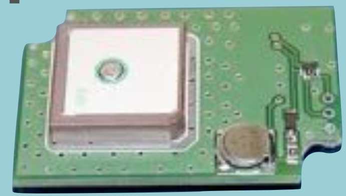
START **%0101 0101** **STOP**

0x55

DECODING HEX 55 (DEC 72)



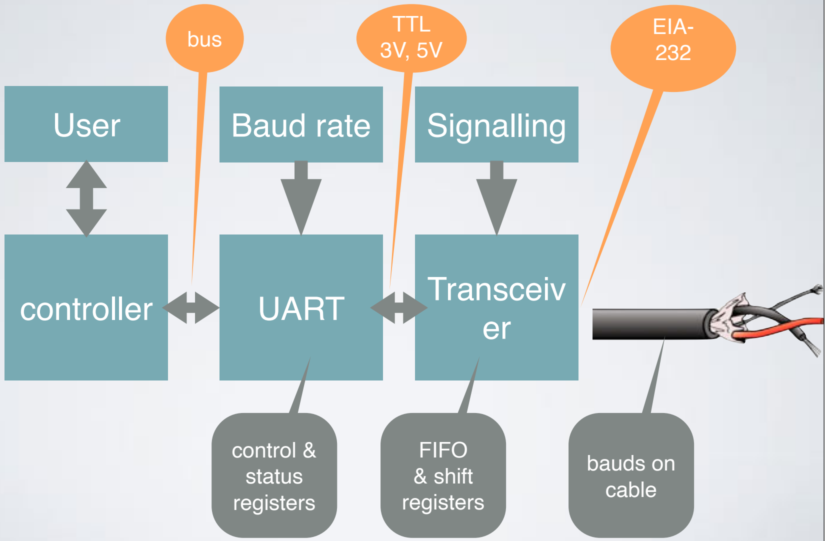
UART



Universal Asynchronous Receiver Transmitter

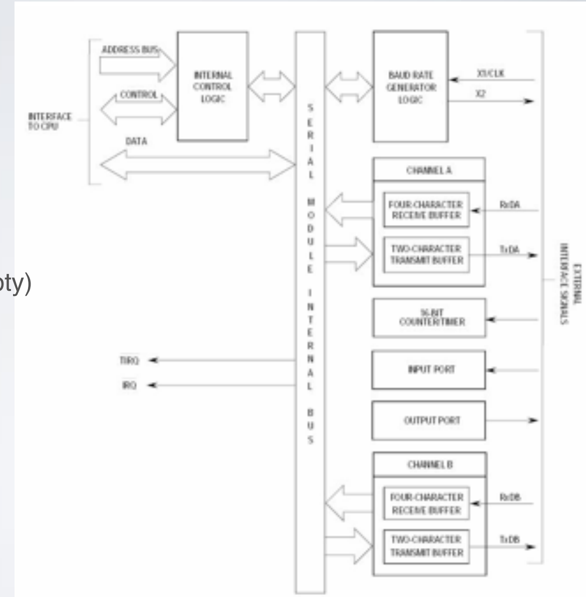
Module 1.2

TRANSMITTER

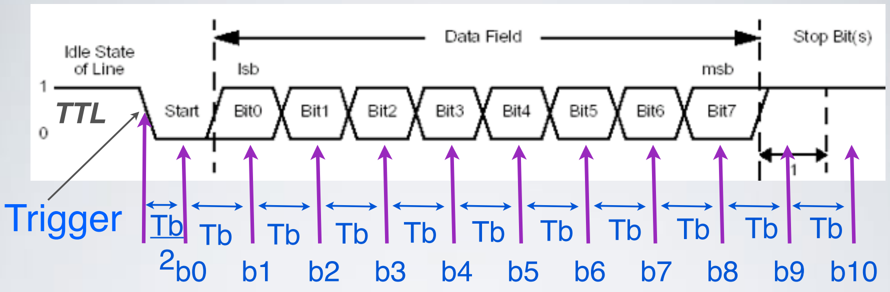


Universal Synchronous/Asynchronous Receiver

- **Tx/Rx:**
  - Data-in register
  - Data-out register
  - FIFO's
- **Status register**:
  - Tx byte sent (Tx_empty)
  - Rx byte ready
  - Rx overrun
  - Framing error
-



RECEIVER TIMING



Tb= 1/ baud rate

0.2 mS for 4800 bps

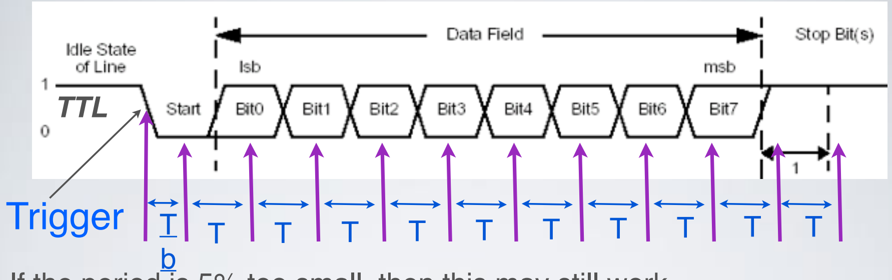
0.1 mS for 9600 bps

4 μS for 250 kbps

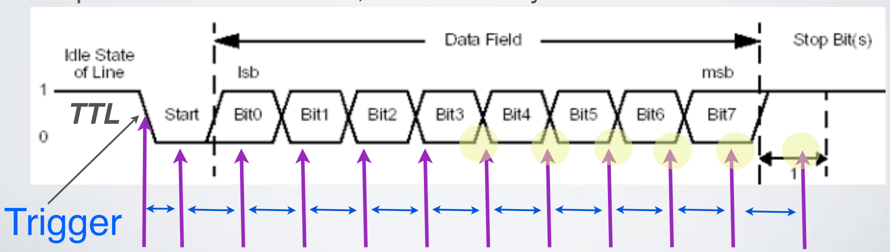
b0 (start) must be “0”; b9, b10 must be “1” -- any other value is an error

Some systems use “parity” by enforcing a check on the last bit before the stop bit (usually uses 9 data bits)

RECEIVER MIS-TIMING

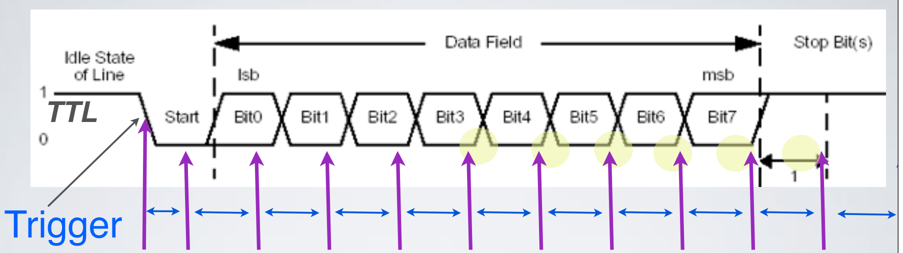


If the period is 5% too small, then this may still work



If the period is significantly too small, then this will fail

RECEIVER MIS-TIMING



If the period is too large, then this will also fail

SUMMARY: ASYNCHRONOUS

Benefits

One common standard (widely supported)

Simple UART implementation, no clock recovery, no DPLL

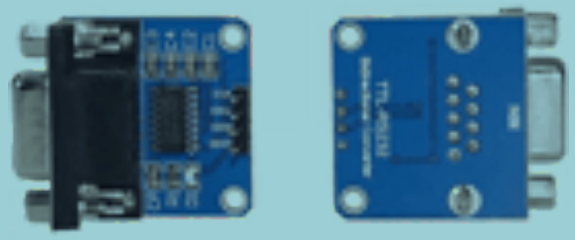
Drawbacks

Lower efficiency: 3/11 of capacity used for framing

Poor error detection, bytes/slots may be “lost”

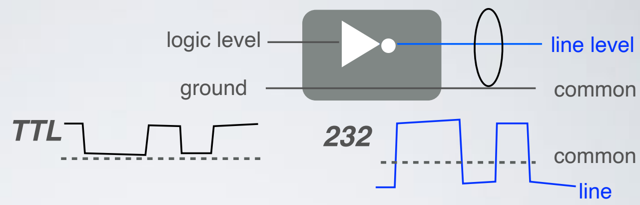
Rate limited by clock stability and cable quality, distance, etc.

EIA-232 TRANSCEIVER



Module 1.3

EIA-232 SERIAL INTERFACE



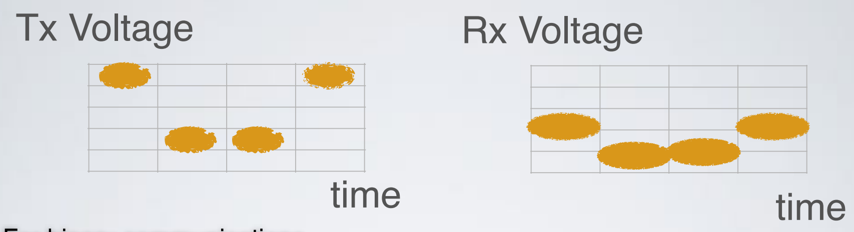
A line transceiver converts logical level signals to bauds

Each baud is sent as a level relative to the common (ground)

A '0' is sent as +12V (**relative to ground**)

A '1' is sent as -12V (**relative to ground**)

BINARY COMMUNICATIONS



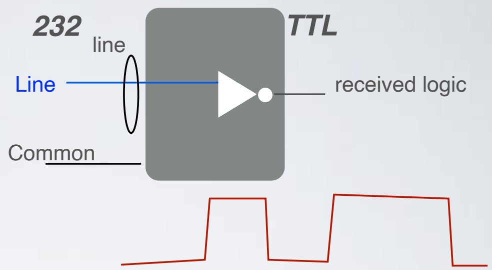
For binary communications

- Receiver needs ***sufficient voltage*** to differentiate a 1 and 0 baud
  - Cable attenuation (resistance/metre) reduces received signal
  - If a 0 is detected when 1 was sent, or vice versa, there is an ***Error***

The cable can be screened at the sender to reduce interference

Reliably drive cables unto 15 metres at 20 kbps or 150m at 9600 bps

EIA-232 RECEIVER



Receiver

Input Impedance 12K Ohms

Shield not connected at the receiver

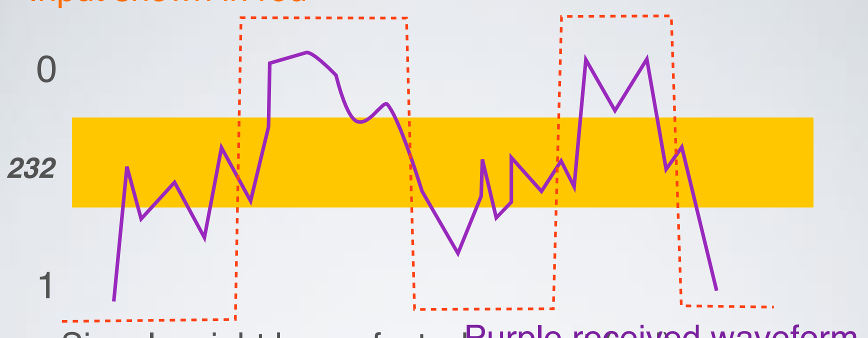
Receiver detects data by reference to the common signal*

Logic 0 when received voltage is between +3V and +15V

Logic 1 when received voltage is between -3V and -15V

** +/- 5V transmitter would also drive a receiver over shorter lengths of cable*

SIGNAL AT THE SENDER



- Signals might be perfect when sent (red)
  - Cable, Receiver and Noise add to distort the waveform
  - Interference from other signals add to the received signal

SIGNAL AT THE RECEIVER

- Received signal often far from perfect
  - Attenuation from the cable - increases with distance
- A large (24V) transmit voltage swing
  - Needed for enough signal at the end of a long cable!

RECEIVER HYSTERESIS

- Receiver uses a voltage threshold hysteresis to avoid oscillation in the output when the signal varies.
  - A signal must cross ***upper*** threshold to count as a 0
  - A signal must cross ***lower*** threshold to count as a 1
  - Prevents rapid transition but causes timing ***jitter***

EIA-423 TRANSMISSION

EIA-423 is an update to EIA-232 for use in an office

Small signals allow higher speeds of 100 kbps

Signal relative to ground (+4 to +6V and -4 to -6V)
Receiver uses a +3V threshold
The signal has a 10V swing (compared to 24V for EIA-232)

Open-ended cable length also increased to 1200m

However, this is not suited to industrial use
- because it is sensitive to noise and interference

SCOPES

10 GBPS 6G LINK

- Prototype 6G subTHz link on display at the 6G Symposium, University of Surrey, May 2023.

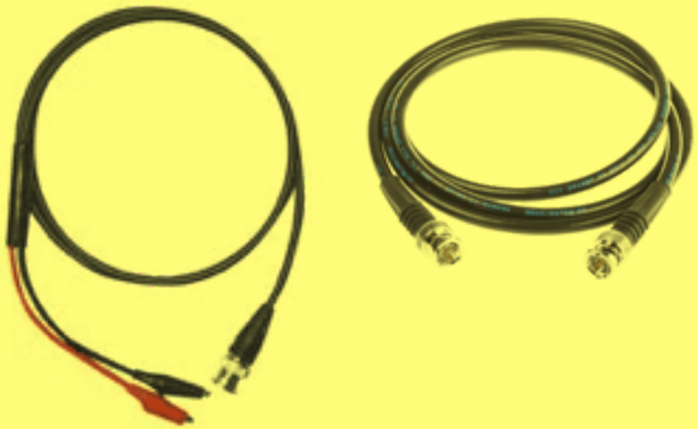
PROBES

PROBES

PROBES

PROBES

PROBES:



PROBES:

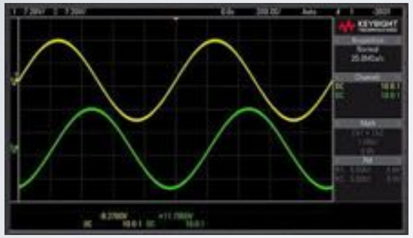
PROBE TIPS

: Check the probe - especially for measurements >20MHz
: Check the coupling mode
: Check the ground connection
: Check the scope channel display matches the probe type!

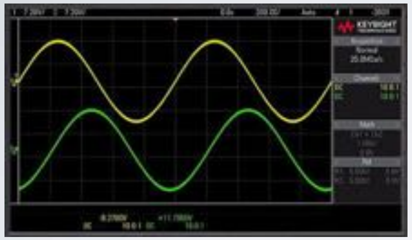
1:10 Impedance Ratio

1:1 Impedance Ratio

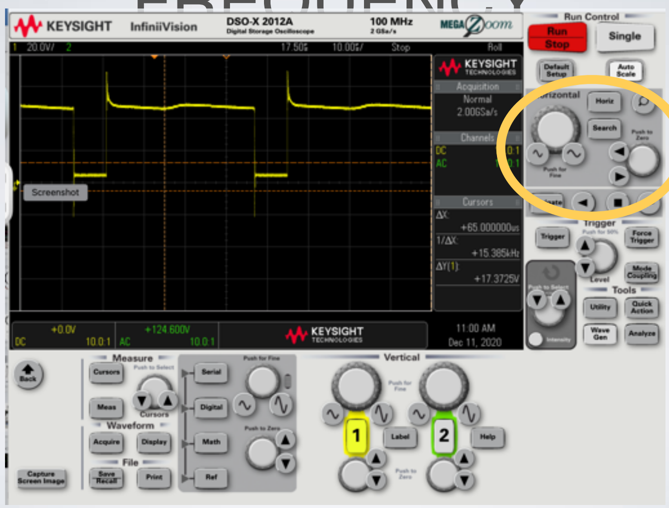
TIMEBASES AND HORIZONTAL CONTROLS



TIMEBASES AND HORIZONTAL CONTROLS



TIMEBASES AND FREQUENCY



CURRENT PROBE

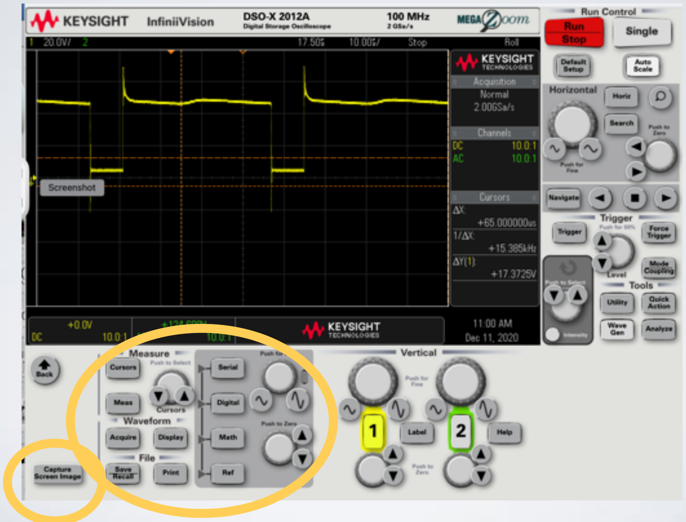
Turn OFF the probe when not in use!!!

For more information about different probes see also: www.keysight.com/find/probes

CURRENT PROBE

CURRENT PROBE

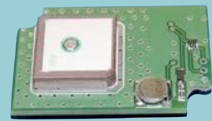
WEB-INTERFACE



COMMUNICATIONS LINKS

Module 2.0

ASYNCH SERIAL FRAMES



GPS NMEA Protocol

Plug & Play ... and very easy to program

EIA-232 interface (up to about 15m)

Low-speed asynchronous bus at 4800 bps

Uses ASCII framed messages

Module 2.1

GPS RECEPTION

- **NMEA standard (National Marine Electronics Association)**
  - A combined electrical and data specification for communication between marine electronic devices
  - Example uses: echo sounder, sonars, anemometer, gyrocompass, autopilot, GPS receivers and other instruments
- **Uses a simplex (unidirectional link)**
  - Sender transmits frames of ASCII characters using a serial link.
  - One sender, but could be one or multiple receivers
-

GPS DATA FORMAT

- Interface: EIA-232/EIA-432 or TTL
  - Serial format, 4800 baud, 8-bits, 1-stop-baud, no parity
  - More on this in the next set of slides...
- Simple frame: starts with a fixed w[illegible]n marker sequence
  - $GPsxx ,,,,
  - Values are represented inASCII

FRAME SYNCH

- Data is grouped into **frames**
  - This allows a receiver to make sense of received data
- A method is needed to align to the start of each frame
  - A sequence may be sent within the data of a frame in a ***Frame Alignment Word*** -typically at the start of each frame.
  - This could also be a ***distinct signal*** at the physical layer.

NMEA DATA FRAMES

- *GGA —Global Positioning System Fixed Data*
  $GPGGA,161229.487,3723.2475,N,12158.3416,W,1,07,1.0,9.0,M,0000*18
- *GLL—Geographic Position - Latitude/Longitude*
  $GPGLL,3723.2475,N,12158.3416,W,161229.487,A,A*41
- *GSA—GNSS DOP and Active Satellites*
  $GPGSA,A,3,07,02,26,27,09,04,15, , , , , ,1.8,1.0,1.5*33

NMEA FRAME SYNCH

- One simple frame: uses a fixed well-known marker field in the first 3 bytes of each frame:
  - $GP.....
  - $GP.....
  - $GP.....

- Any unexpected values result in the entire frame being discarded, and the receiver has to hunt for synchronisation.

FRAME ALIGNMENT

- First stage, search for the $GP pattern….

19,4807.038,N,01131.000,E1,08,0.9,545.4
,M,46.9,M,,?$GPGGA,123519,4807.038,N011
31.000,E,1,08,0.9,545.4,M,46.9,M,,?$GPG
GA,123519,4807.038,N,01131.000,E,1,08,0
.9,545.4,M,46.9,M,,?$GPGGA,123519,4807.
038,N,01131.000,E,1,08,0.9,545.4,M,46.9
,M,,?$GPGGA,123519,4807.038,N,01131.000
,E,1,08,0.9,545.4,M,46.9,M,,?

TRANSMISSION ERROR

What happens when bits are corrupted by noise?

19,4807.038,N,01131.000,E1,08,0.9,545.4
,M,46.9,M,,?$GIPGGA,123519,4807.038,N011
31.000,E,1,08,0.9,545.4,M,46.9,M,,?$GPG
GA,123519,4807.038,N,01131.000,E,1,08,0
.9,545.4,M,46.9,M,,?$GPGGA,123519,4807.
038,N,01131.000,E,1,08,0.9,745.4,M,46.9
,M,,?$GPGGA,123519,4807.038,N,01131.000
,E,1,08,0.9,545.4,M,46.9,M,,?

Error, needs to be detected

CHARACTER PARITY

0	1	2	3	4	5	6	7	EVEN
1	1	1	0	0	0	1	0	0
0	0	0	1	0	0	1	0	0
1	0	0	1	0	0	1	0	1
1	0	0	1	0	0	1	0	0

- Parity baud **sent** as XOR of 8 data bauds
  - *Number of 1 bits + parity always an even number of 1's*
- Parity **checked** as XOR of 8 data bauds = Parity baud
- If parity is incorrect, byte is marked as an error (red

INTEGRITY CHECK AT END

- Sample: $GPGGA,123519,4807.038,N,01131.000,E,1,08,0.9,545.4,M,46.9,M,,***47**

Final byte in a frame can contain a binary number to check frame inetgrity
(here written here as * and two hex digits)

Cumulative XOR of all bytes between the $ to the *.
(also known as longitudinal parity)

```
var checksum = 0;
for(var i = 0; i < stringToCalculateTheChecksumOver.length; i++) {
  checksum = checksum ^
    stringToCalculateTheChecksumOver.charCodeAt(i);
}
```

LONGITUDINAL PARITY

Receivers compare transmitted parity in the message with a value re-calculated at the receiver.

- Longitudinal parity for $GP
  - sent parity "00110011"
  - received "00110011"
- Sent=Received parity - ***OK***!

$	G	P	PARITY
0	0	0	0
0	1	1	0
1	0	0	1
0	0	1	1
0	0	0	0
1	1	0	0
0	1	0	1
0	1	0	1

LONGITUDINAL PARITY

- Longitudinal parity for $GP
  - sent parity “00110011”
  - received “00111011”
- Sent ≠ Received
  - One ***error*** detected!

$	0	P	PARITY
0	0	0	0
0	1	1	0
1	0	0	1
0	0	1	1
0	1	0	0
1	1	0	0
0	1	0	1
0	1	0	1

LONGITUDINAL PARITY

- Longitudinal parity for $GP
  - sent parity “00110011”
  - received “001110001”
- Sent ≠ Received
  - Multiple errors detected!

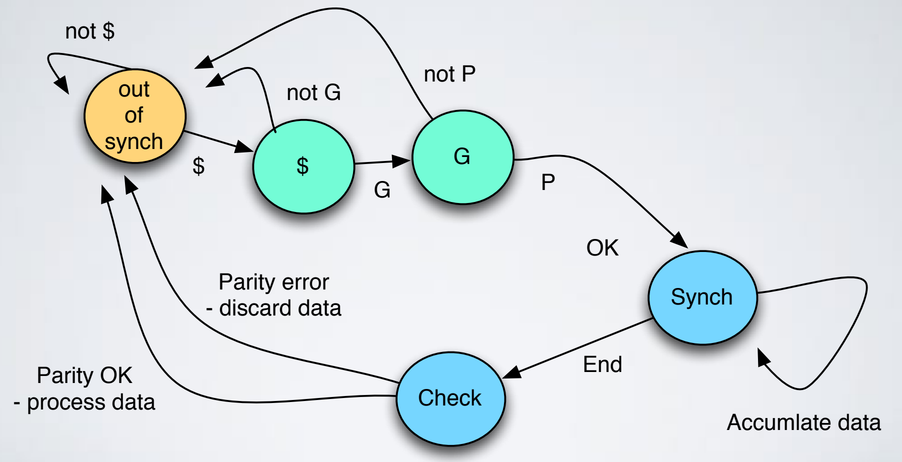
$	0	P	PARITY
0	0	0	0
0	1	1	0
1	0	0	1
0	0	1	1
0	1	0	0
1	1	0	0
1	1	0	1
0	1	0	1

LONGITUDINAL PARITY

- Longitudinal parity for $GP
  - sent value “00110011”
  - received “00110011”
- Even errors ***NOT*** detected!

$	0	P	PARITY
0	0	0	0
0	1	1	0
1	0	0	1
0	0	1	1
1	1	0	0
1	1	0	0
0	1	0	1
0	1	0	1

RECEIVER STATE



Note any error also causes return to the out-of-synch state

IMPROVING FRAME ALIGNMENT

- NMEA GPS sends a ***continuous stream*** of updated messages
  - Framing relies on a unique '$' character not intentionally appearing in data.
  - Corrupted data is discarded, there is no retransmission - - receiver simply waits for next updated message.
- Doing better:
  - Could be ***robust*** to corruption of frame alignment word - i.e. a corruption does not cause immediate loss of synchronisation.
  - Most NMEA systems use ***differential transmission*** (see next module)

CAREFULLY WRITE ABOUT YOUR RESULTS

How accurate is the measurement?
If we look for a cat, it is either there or not...
If we have a picture that shows a cat, there may be doubt?

Examine the accuracy:
- How accurately can you really measure?
- How repeatable is the result?

Be careful about describing your results:
- What did you measure? (what units??)
- How many figures of accuracy should you cite?
- Are your results within a referenced norm for the measurement?

BE MINDFUL OF THE ORIGINS OF IDEAS

The more we focus on our ideas in a way that systematically ignores their objective origins, the more unreliable those ideas become...

Examine our sources
- How do we know our facts are trusted? who says so?

Provide evidence at multiple levels:
- Primary sources - Published International Standards
- Secondary sources - Reviewed papers, Books, etc (explanation...)
- Supporting sources - product data; web pages; etc (how..

EXAMPLES

If you measure the baud rate as 9601 bps
- What is the expected nominal rate?
- How accurately can you measure?
- Is this variation acceptable

If you measure 12.001 volts
- What is the expected nominal rate?
- How accurately can you measure?
- Is this variation acceptable

Take care in how you make your conclusions./

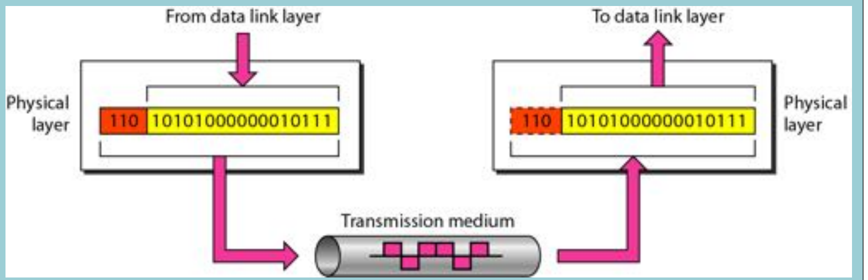
ASSIGN YOUR MARK

Marking Checklist

EIA-485 DIFFERENTIAL ASYNCHRONOUS SERIAL EQUIPMENT BUS

Module 3.0

TRANSMISSION THEORY

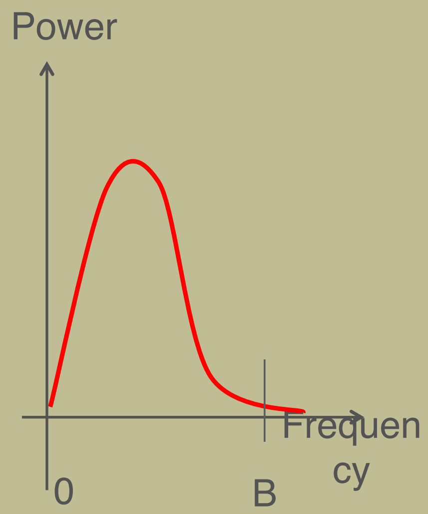


Each bit is sent as a discrete signal (baud) along the wire.

The transmission medium can be considered a “channel”

Module 2.2 (May in some years be presented as a part of Module 1)

IDEAL NOISE-LESS

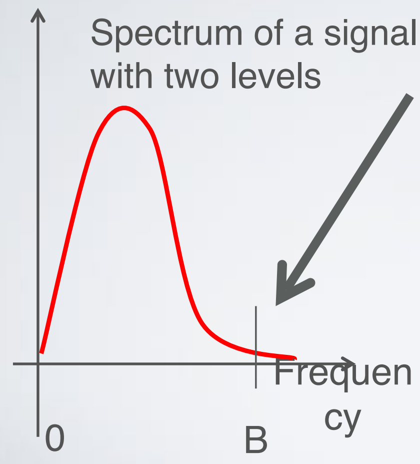


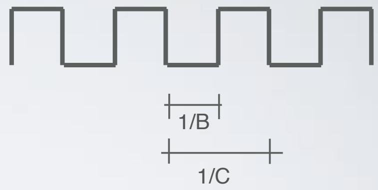
H. Nyquist

NYQUIST FREQUENCY

- Consider an ideal channel (no noise)
  - The sender transmits two levels (“0” or “1”)
- Maximum transmission rate of a signal over a cable with fixed bandwidth
- Transmission capacity (C) is twice bandwidth (B):
  - C = 2 x B

EXPLAINING THE NYQUIST THEOREM

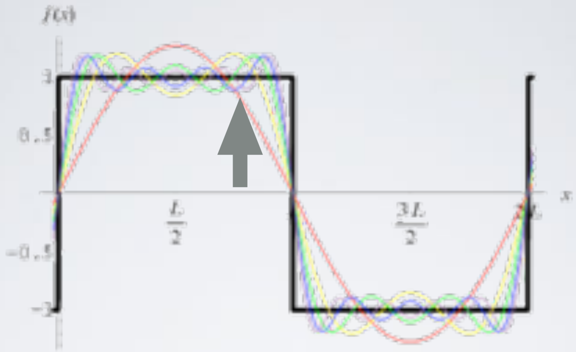




- Largest frequency component of signal is when the signal alternates high/low:

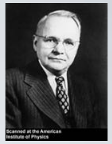
2/B = 1/C

FOURIER DECOMPOSITION

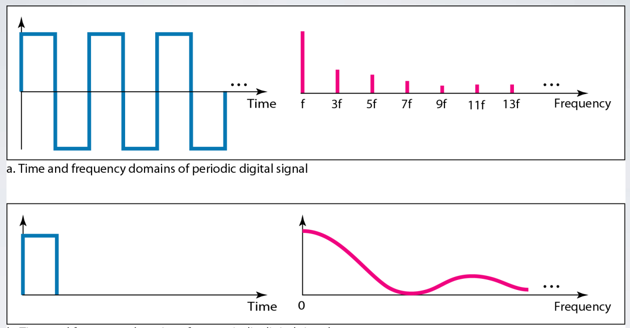


Fourier analysis can decompose a periodic signal into a combination of sine waves with different frequencies, amplitudes, and phases.

FOURIER DECOMPOSITION



H. Nyquist



a. Time and frequency domains of periodic digital signal

b. Time and frequency domains of nonperiodic digital signal

A perfect digital signal has an infinite bandwidth….

Note for later:
Real cables have resistance - attenuation/metre
And capacitance/inductance - limiting cable bandwidth

FILTERING HIGHER ORDER HARMONICS

H. Nyquist

- Filtering higher order harmonics result in a smoother signal
  - A receiver needs to sample at the centre of a baud to detect the level (0 or 1)
- Nyquist filtering limits the signal spectrum bandwidth(0Hz to B)
  - Nyquist theorem would require the spectrum to be exactly zero when frequency>B

SIGNAL RATE

What is the required bandwidth of a low-pass channel if we need to send 1 Mbps using baseband transmission?

Solution

a. Minimum bandwidth, B = bit rate /2, or 500 kHz.

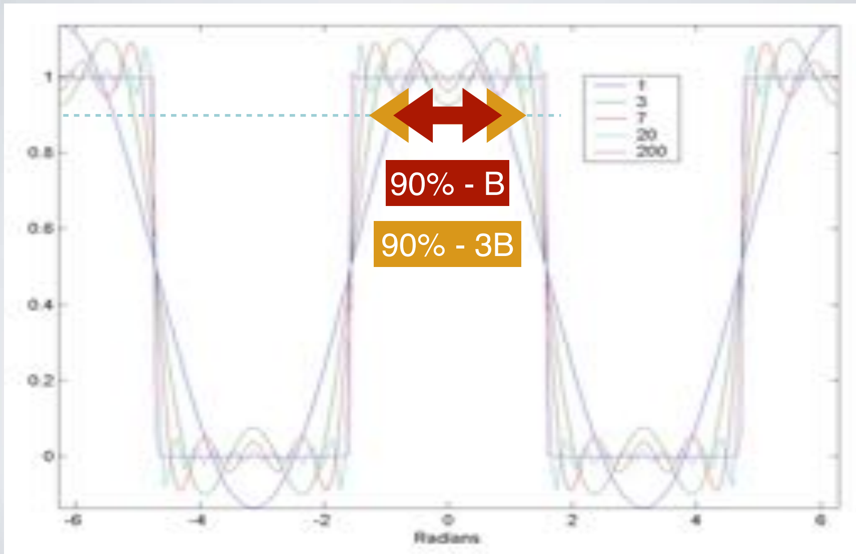
SIGNAL BANDWIDTH

- What is the required bandwidth of a low-pass channel if we need to send 1 Mbps using baseband transmission?
- Solution
  The answer depends on receiver.
  a. The minimum bandwidth, is B = bit rate /2, or 500 kHz.

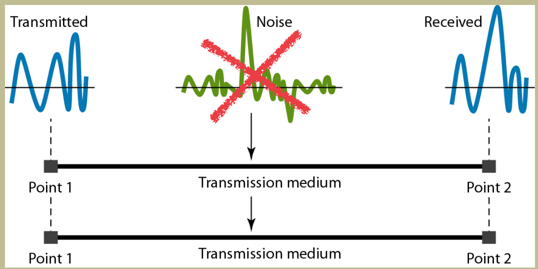
  b. A more "square" waveform eases receiver timing ..
  e.g. to include the first and the third harmonic harmonics with B = 3 × 500 kHz = 1.5 MHz.

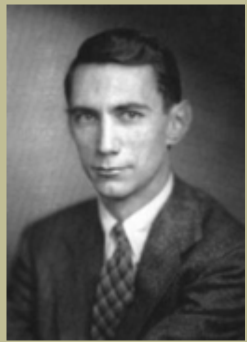
  The first, third, and fifth harmonics would be:
  B = 5 × 500 kHz = 2.5 MHz.

SAMPLING POINT



A REAL "CHANNEL"

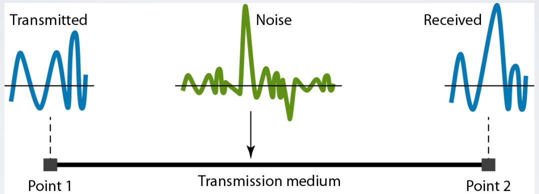




C. E. Shannon

NOISE

Real channels have limits…

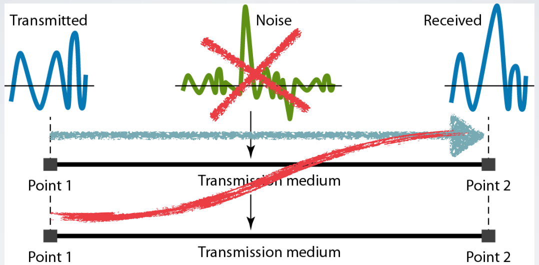


There is no such thing as a noiseless channel!!

INTERFERENCE

Industrial environments can be hostile - our signal is not alone

Noise+Interference



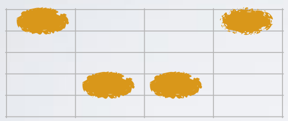
Other signals can also be received*, increasing the noise floor

Far-end cross talk is a measure of the received unwanted signal

* Later in the course we'll see that similar signals can have the same frequency spectrum and are particularly disruptive

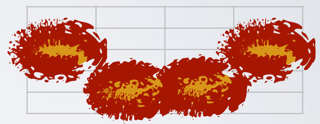
BINARY COMMUNICATIONS WITH NOISE

Tx Voltage



time

Rx Voltage

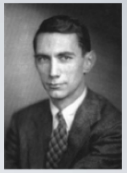


time

Received signal is not just smaller due to length of cable (attenuation)

Noise is also added to the cable signal and at the receiver

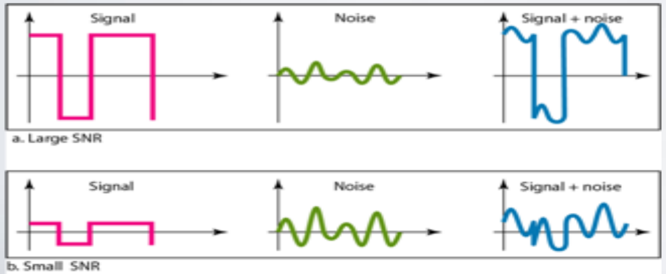
A REAL "CHANNEL"



C. E. Shannon

- Noise and interference make small signals difficult to detect
- The important factor is the signal to noise ratio (SNR).

$$SNR = \frac{\text{Power signal}}{\text{Power noise}}$$



High SNR

Low SNR

A REAL WAVEFORM (EYE DIAGRAM)

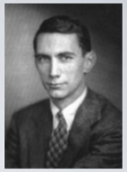
One way to view the signal is an eye diagram

- Scope triggered at a particular point (start of a baud)
- Each trigger, scope resets the X-axis
- It does not erase the display (persists for multiple scans)

High SNR
(e.g., at sender)

Lower SNR
(e.g., at receiver)

SHANNON CAPACITY



C. E. Shannon

- For a noisy channel, the Shannon capacity gives a theoretical limit of the usable bitrate of a channel with a bandwidth B and a signal-to-noise-ratio SNR.

  C = B x log2(1 + SNR)

- Any attempt to transmit faster than the Shannon limit will result in unrecoverable transmission errors

THEORETICAL CAPACITY OF A TELEPHONE LINE

- A telephone line has a nominal bandwidth of 3000 Hz and the signal-to-noise ratio is 3000 (69.5 dB).
- What is the channel capacity?
- Using Shannon formula, the highest rate is:

  C = 3000 x log2(1 + 3000) = 34.7 kbps.

- If we wish to send faster than, we can either increase the ***bandwidth of the line*** or ***improve signal-to-noise ratio.***

THERE IS A MINIMUM SNR

- Consider an extremely noisy channel with a signal-to-noise ratio of almost zero. i.e. noise so strong that the signal is faint.
  - The signal-to-noise-ratio is very small SNR<<1
- Capacity of a channel tends to zero regardless of the bandwidth:

$$C = B \log_2 (1 + \text{SNR}) = B \log_2 (1 + 0) = B \log_2 1 = B \times 0 = 0$$

LABS

A: EIA-232

B: ASYNCHRONOUS COMMS

Lab Notes

CAREFULLY WRITE ABOUT YOUR RESULTS

How accurate is the measurement?
If we look for a cat, it is either there or not...
If we have a picture that shows a cat, there may be doubt?

Examine the accuracy:
- How accurately can you really measure?
- How repeatable is the result?

Be careful about describing your results:
- What did you measure? (what units??)
- How many figures of accuracy should you cite?
- Are your results within a referenced norm for the measurement?

BE MINDFUL OF THE ORIGINS OF IDEAS

The more we focus on our ideas in a way that systematically ignores their objective origins, the more unreliable those ideas become...

Examine your sources
- How do we know our facts are trusted? who says so?

Provide evidence at multiple levels:
- Primary sources - Published International Standards
- Secondary sources - Reviewed papers, Books, etc (explanation...)
- Supporting sources - product data; web pages; etc (how...)

EXAMPLES

If you measure the baud rate as 9601 bps
- What is the expected nominal rate?
- How accurately can you measure?
- Is this variation acceptable

If you measure 12.001 volts
- What is the expected nominal rate?
- How accurately can you measure?
- Is this variation acceptable

Take care in how you make your conclusions.!!!!

ASSIGN YOUR MARK

Marking Checklist

EIA-485 DIFFERENTIAL ASYNCHRONOUS SERIAL EQUIPMENT BUS

Module 3.0

DIFFERENTIAL TRANSMISSION

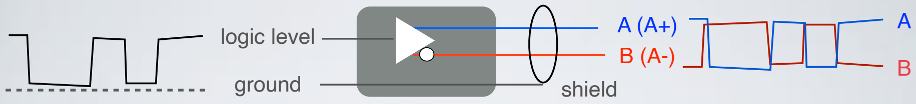
INTERFACE

Module 3.1

EIA-485 TRANSMISSION

1. Differential transmission
2. Balanced cable pair
3. Multi-drop bus - one sender, multiple receiver

EIA-485 TRANSMITER



A line transceiver converts **logical level signals to line levels**

The output sends the signal using two conductors A and B*

The difference between A and B is ***always*** 5V

The cable shield/screen is grounded only at the sender

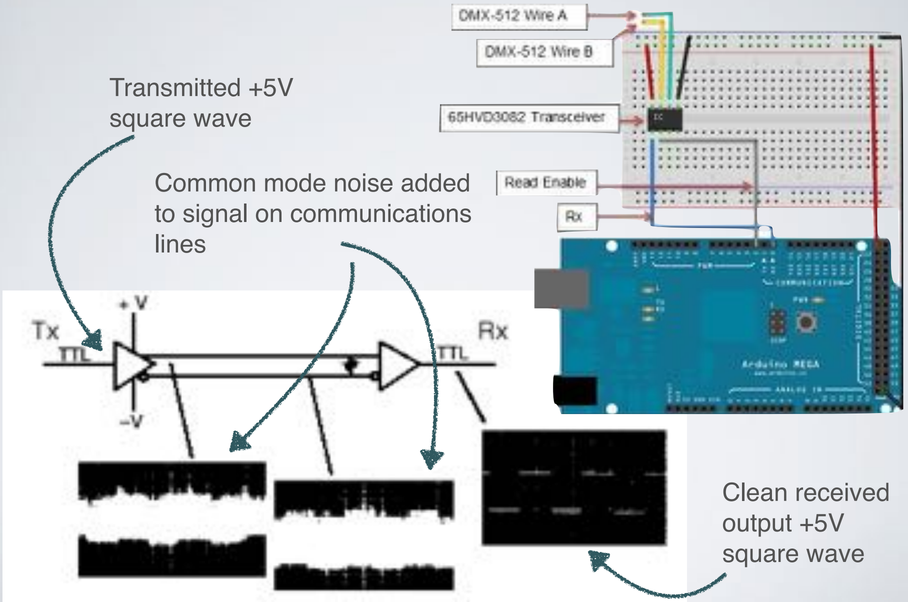
Each baud is sent by setting the level of A and B:

The B signal is an inverted A signal (there is no average dc voltage)

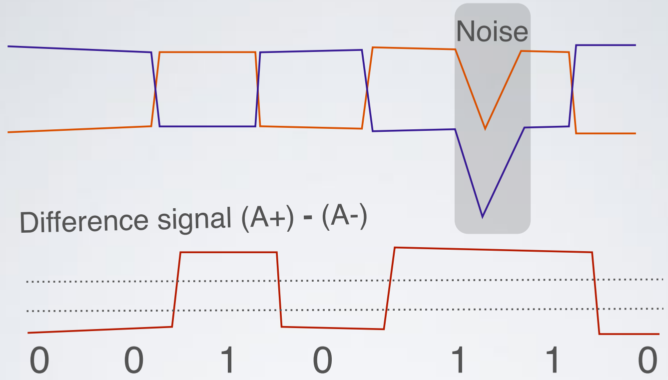
Reliably drives cables unto 1000 metres at 250 kbps

** The B signal is also known as "A-"*

SIMPLE TRANSCEIVER



DIFFERENTIAL

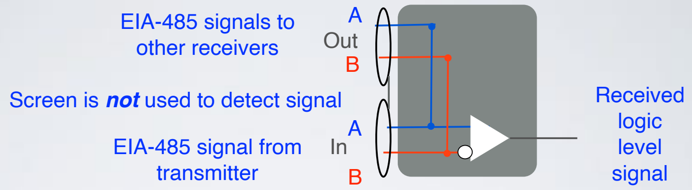


Common-mode noise impacts both conductors equally

There is no net difference form common-mode noise

Significant increase in noise immunity

EIA-485 RECEIVER



A receiver detects data by the ***difference*** between the two conductors

This uses a 200mV Differential threshold detector

Logic 0 a difference between (A+ & A-) < 200mV

Logic 1 a difference between(A+ & A-) > 200mV

The A,B signals are not referenced to the cable screen

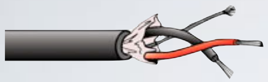
Can be in the range + 12V, -7V relative to the receiver ground

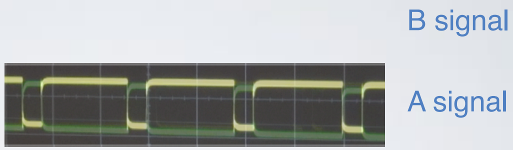
DIFFERENTIAL TRANSMISSION

CABLE

Module 3.1

EIA-485 TRANSMISSION





Balanced cable improves noise immunity
Differential signal, 200mV receive threshold

Two advantages:
Sending signal can be small (+5V differential signal
- Allows ***higher speed***
Received signal can be small (High impedance 12K Ohm)
- Allows ***multiple receivers*** with one sender

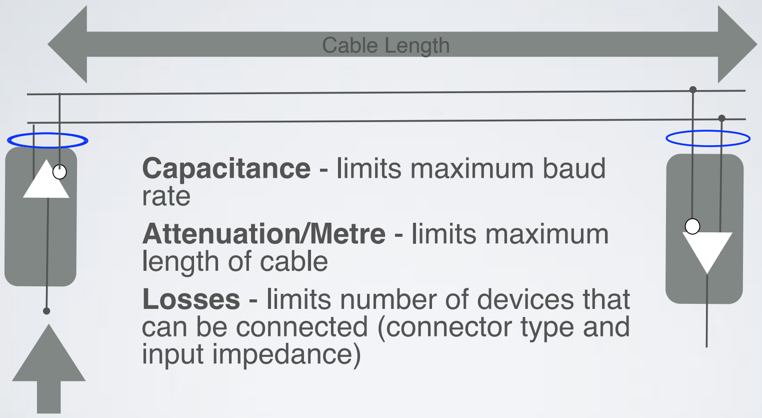
Well suited to industrial use
- insensitive to noise and interference

EIA-485 CABLE

2 stranded conductors twisted along the length forming a “pair”
Impedance between the pair: 100-120 Ohms
Conductive shield around the twisted pair (ground) - ***see later***
PVC outer protective sleeve

Capacitance between conductors within a shield < 65 pF/m
Capacitance between any conductor and the shield < 115 pF/m

CABLE CHOICE



Capacitance - limits maximum baud rate

Attenuation/Metre - limits maximum length of cable

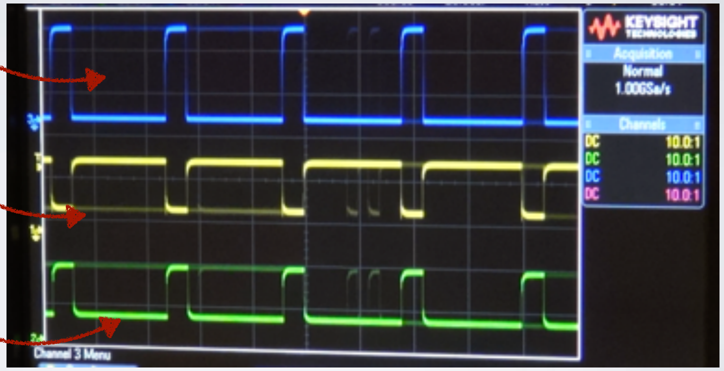
Losses - limits number of devices that can be connected (connector type and input impedance)

LINE SIGNALS & TRANSCEIVER OUTPUT

output signal (blue)

inverted B signal

non-inverted A signal



EXAMPLE CABLE

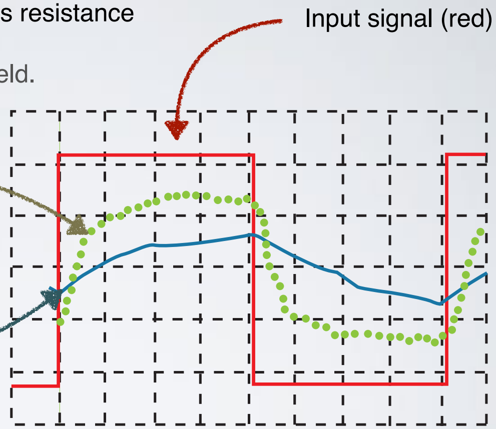
Cable has capacitance as well as resistance

typ. 30-60 pF conductor to shield.

The cable acts as a low pass RC filter.
<1 Mbaud ~ 1 MHz signal
Higher frequencies > MHz attenuated (green)!

higher frequencies e.g. 30 MHz (Blue) are severely attenuated

Input signal (red)



A signal working at 300kbaud is not distorted.

Maximum rate determined by capacitance & resistance

HOW MUCH SIGNAL IS RECEIVED?

Signal transmitted at sender 5V

Cable attenuation and loss reduce the signal level (~ 4db/100m)

Minimum signal at receiver 0.2V

Let's calculate what that means for a practical system with:

300m of cable

32 receivers

CABLE POWER MARGIN*

Signal transmitted at sender 5V

Minimum signal at receiver 0.2V

Power margin in decibels

$= 10 \log (V_{in}/V_{out})^2$

$= 20 \log (V_{in}/V_{out})$

$= 20 \log (5/0.2)$

$= 28$ dB

The receiver signal can be 28dB lower than the sender

*Power margin is measured in dB

CABLE ATTENUATION 24/7 TWISTED PAIR

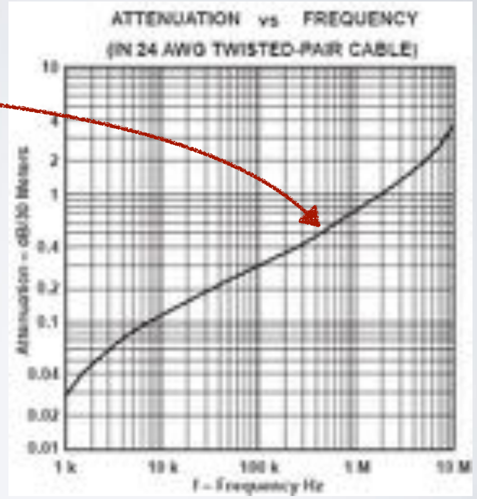
Resistance: 85 Ohm/km

Typical attenuation is:
~2-4dB per 100m
@4dB/100m:
For total cable bus length:
300m = ~12dB

For each receiver:
0.1dB loss per transceiver
0.2dB connector loss
Total loss /receiver 0.3dB
Loss from 32 receivers = 13 dB

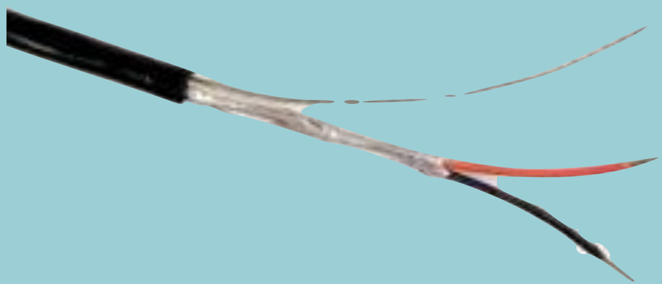
Signal attenuation at end = 25dB

Margin = 3dB



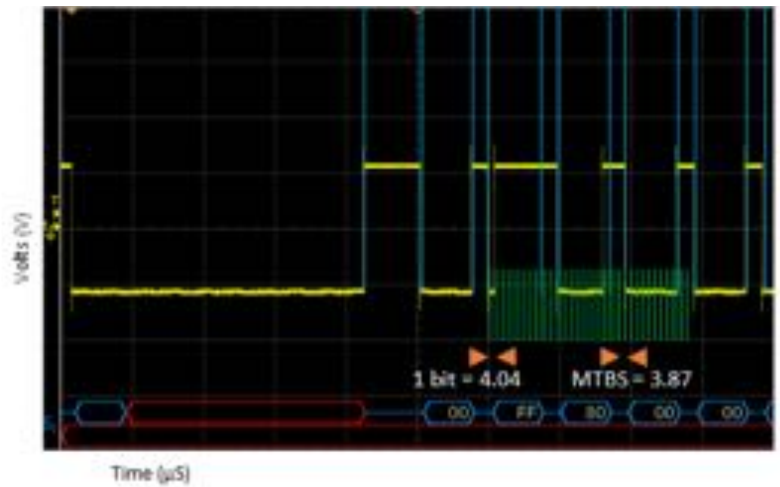
Maximum distance was limited by number of receivers & cable length

BUS TRANSCEIVERS



Module 4.3

DMX Frame Timing on Bus

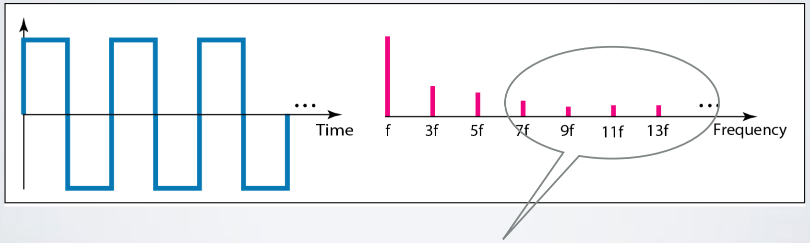


SIGNALS ON CABLE

A square wave contains frequency harmonics many times baud rate.

For 250 kBaud, highest frequency components arise sending 101010 etc

- Highest rate => 125 kHz square wave
- Components at 375 kHz, etc ….



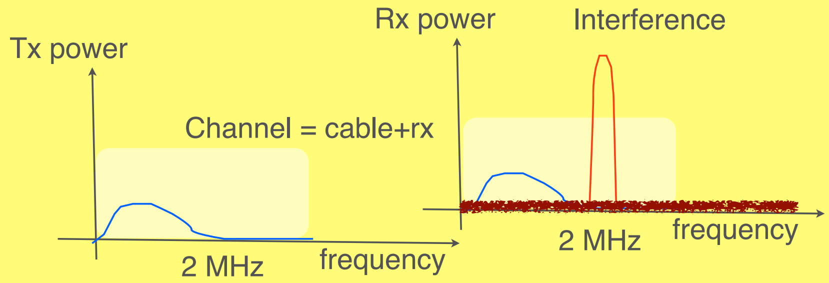
There is still appreciable energy ***above 2 MHz***

SIGNALS ON CABLE

Signal energy mainly around baud rate (<< 2 MHz)

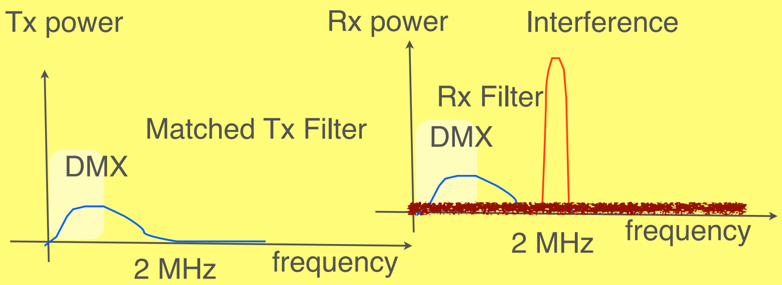
Signal has components >> 2 MHz

Interference/noise above 2 MHz degrades signal!



Effect of high-rate transceiver

TOO LOW SLEW RATE

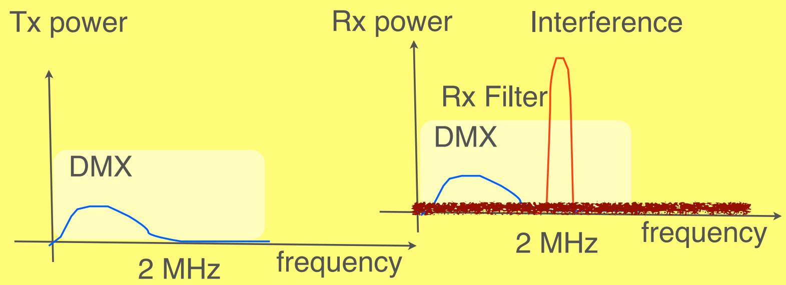


Effect of matched transceiver

A too narrow filter removes some of the wanted signal

- Lowers signal to noise ratio (filters a part of signal in frequency domain)
- In the time domain, this causes some signal energy from an older baud to still be present when the next baud is sent (Inter-symbol-interference).

TOO HIGH SLEW RATE

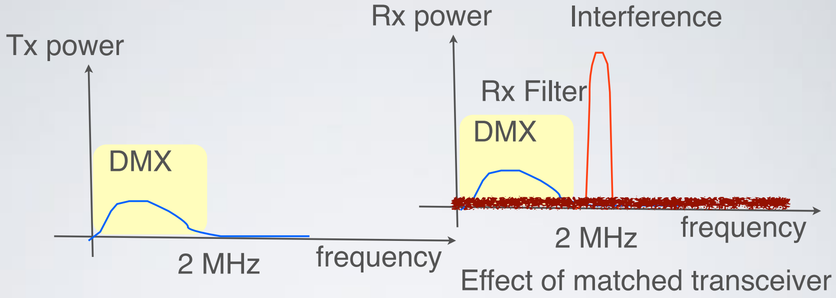


Effect of matched transceiver

Wideband filter fails to effectively remove interference and noise

- lowers signal to noise ratio

LIMITING SLEW RATE



Line drivers use a low-pass filter, shaping signal at sender ***and*** receiver

- This limits slew rate of the signal, or makes the edges "slower"
  - This also increases rise-time of the signal when a level changes
- Half of the filter function is at the sender and half at the receiver
  - Ensures all transmitted energy falls within the receiver filter

SAMPLE AT THE CENTRE!

A shaped signal rolls-off more gently than a digital signal: it becomes important to sample at the centre of each received baud.

WORKING IN HARSH ENVIRONMENTS

Cable

Send more **voltage** to compensate for attenuation/meter
Use ***differential transmission and twisted pair*** cable
Use foil shi***eld, earthed*** at sender
Termination at end of cable to match impedance
Low ***attenuation***/meter

Connection to cable

DC isolation of the bus (removing earth loops)
Eliminate problems from cable breaks (capacitor to ground, input bias)
Avoid ***cable stubs***

Receiver

Limit ***slew rate*** (reduce noise/interference)
Hysteresis (to eliminate effect of transient noise)
Sample at the centre of each baud

EIA-485 EQUIPMENT

Process Field Bus, used mainly in industrial plants (EN 50170).
Field Bus, used for industrial automation.
***CAN Bus**, used for control networks in cars, lifts, etc*
Building automation/management
Common lab/machine room instrumentation bus.

EIA-485 SIMPLEX EQUIPMENT BUS: DMX-512 PHYSICAL LAYER

Module 4.0

DMX-512 OVERVIEW

Multiplexing data using a serial control bus
G Fairhurst

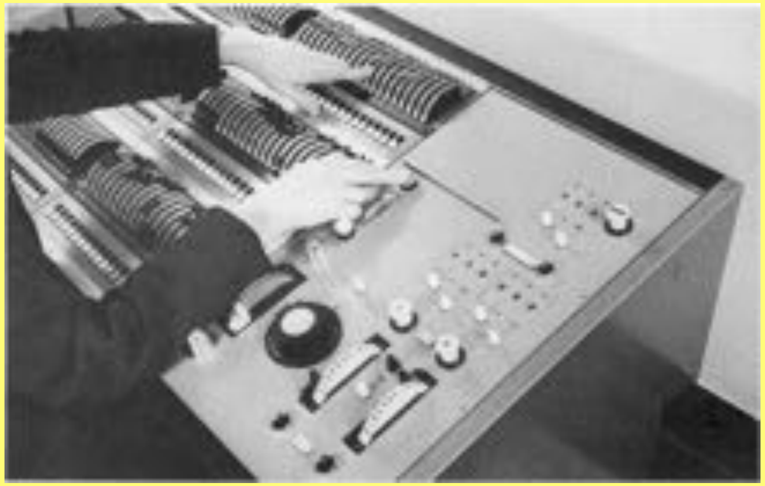
Module 4.1

DMX READING

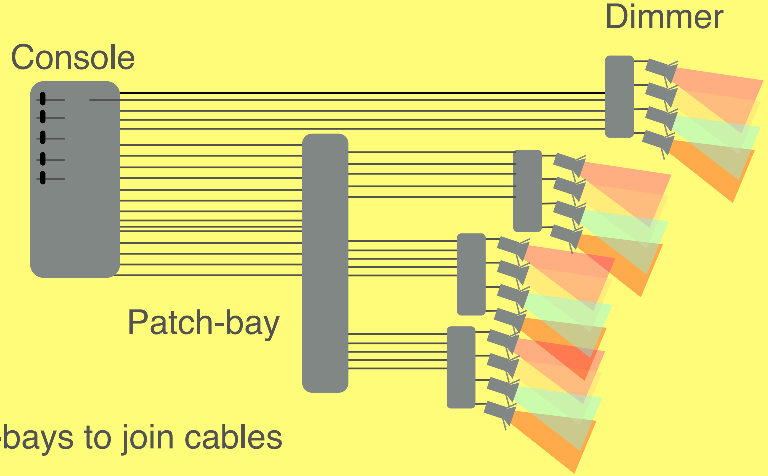
- "Recommended Practice for DMX 512: A Guide For users and Installers", Adam Bennette, (PLASA) *
- “Control Freak - A real world guide to DMX-512 and Remote Device Management”, Wayne Howell, 2010
- ANSI E1.11, Asynchronous Serial Digital Data Transmission Standard for Controlling Lighting Equipment and Accessories, USITT DMX512-A, American National Standards Institute, 1990 (PLASA) *
- ANSI E1.20, Remote Device Management, over USITT DMX 512 Networks, 2003 (PLASA) *

* Free download at tsp.plasa.org

ELECTRICAL DIMMING -



MESSY CABLING



Patch-bays to join cables

LOTS of cable - Lots of connections

DIGITAL MULTIPLEX (DMX)

The DMX-512 standard (actually USITT DMX-512 - 1990)

- Published by U.S.I.T.T. and now maintained by ESTA
- Designed to be easily implemented by microcontrollers

Single simple cable

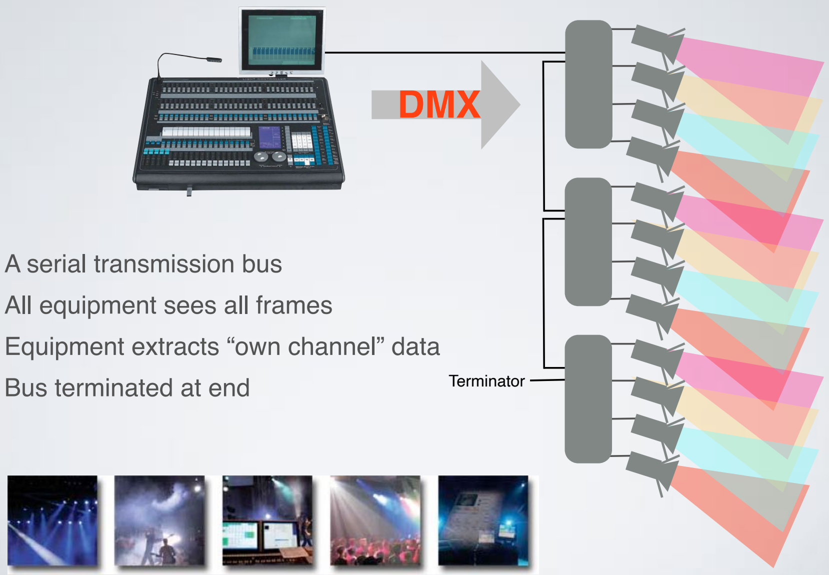
- Assembles channel slots into a 513 slot frame
- One cable is less bulky, cheaper, and less cumbersome
- For long distances, repeaters only need to amplify 2 signals

Standard allows control of a wide variety of equipment:

- PAR cans, moving head lamps, stage equipment, smoke machines, scanners, dimmers, fans, motors, etc.

Equipment may be controlled by more than one channel

DMX BUS



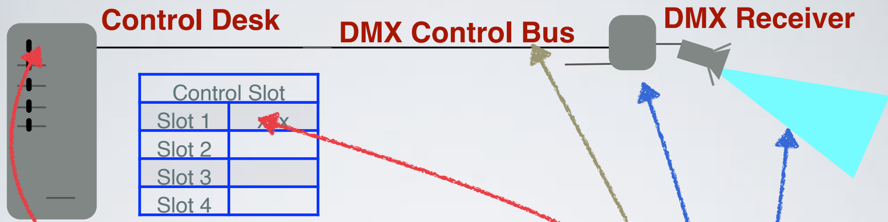
A serial transmission bus

All equipment sees all frames

Equipment extracts "own channel" data

Bus terminated at end

1: DEMONSTRATION SLOT 1



DMX Console sending control data

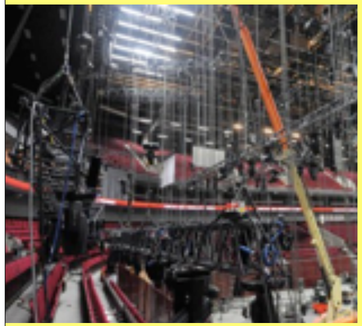
- Console changes value of channel 1
- Value placed in slot 1 of the DMX Frame

DMX Frame sent on the DMX Control Bus

DMX Receiver configured with address 1

- It demultiplexes slot 1 extracting the value
- It sets a corresponding output

EUROVISION 2013



1243 Lighting fixtures

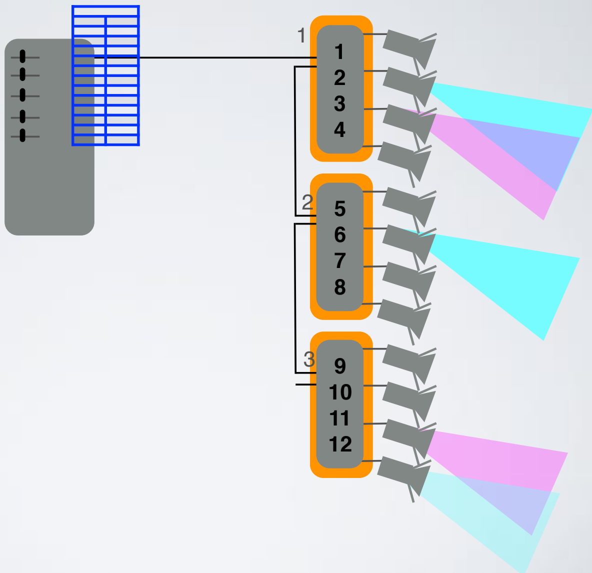
800 moving lights

50 km power cable

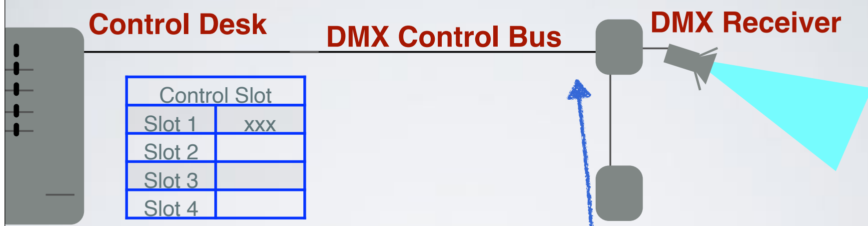
40 km control, video and audio cable

DEMULTIPLEXING

Control Slot	
Slot 1	0
Slot 2	255
Slot 3	128
Slot 4	0
Slot 5	0
Slot 6	255
Slot 7	0
Slot 8	0
Slot 9	0
Slot 10	0
Slot 11	60
Slot 12	20
Slot 13	0



2: DEMONSTRATION TWO RECEIVERS FOR SLOT 1



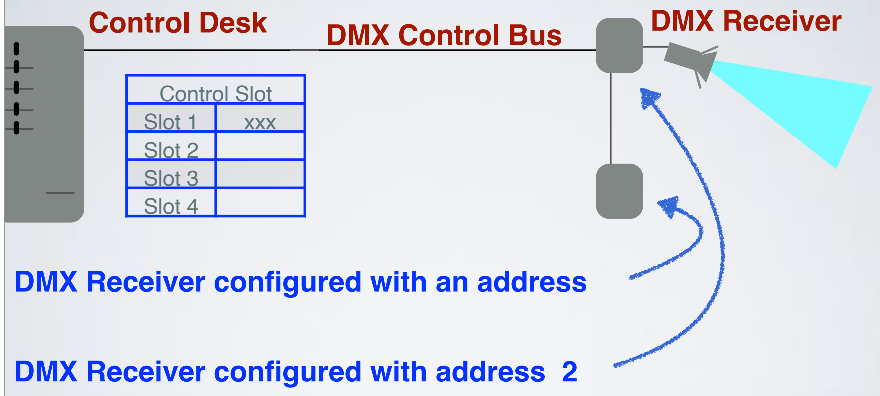
DMX Receiver configured with address 1

DMX Analyser

Receives data from same control bus

Monitors value in slot 1 of the DMX Frame

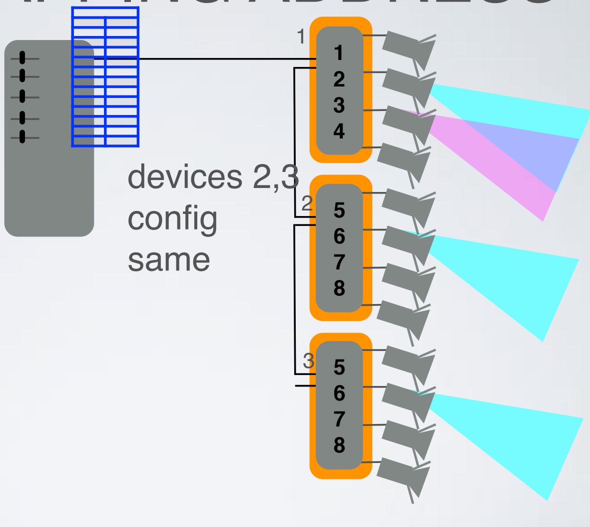
3: DEMONSTRATION DIFFERENT ADDRESSSES



Control Slot	
Slot 1	xxx
Slot 2	
Slot 3	
Slot 4	

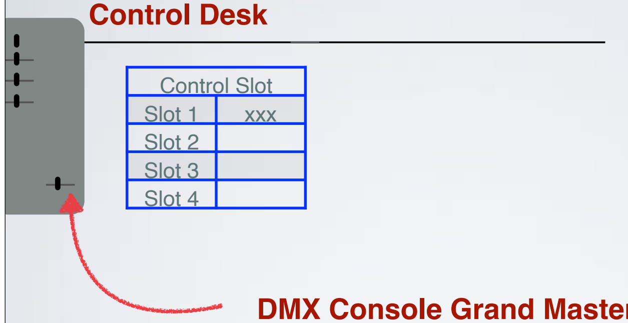
OVERLAPPING ADDRESS

Control Slot	
Slot 1	0
Slot 2	255
Slot 3	128
Slot 4	0
Slot 5	0
Slot 6	255
Slot 7	0
Slot 8	0
Slot 9	0
Slot 10	0
Slot 11	60
Slot 12	20
Slot 13	0



This is a bus - more than one device can read a channel

4: DEMONSTRATION GRAND MASTER



Control Slot	
Slot 1	xxx
Slot 2	
Slot 3	
Slot 4	

DMX Console Grand Master

Scales all the outputs values from the console

This over-rides all the other controls

DMX CONSOLES

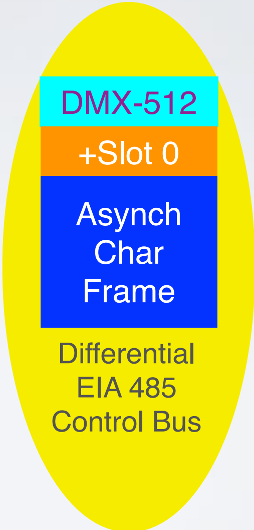
Computer-based

Dedicated Hardware

ASYNCHRONOUS BUS

Logical link framing groups of bytes

Physical transmission of bauds



FRAMES OF SLOTS

- Control Slot
- Slot 1
- Slot 2
- Slot 3
- Slot 4
- ....
- Slot 512

Module 5.1

FRAMES OF SLOTS

Start

- Control Slot
- Slot 1
- Slot 2
- Slot 3
- Slot 4
- ....
- Slot 512

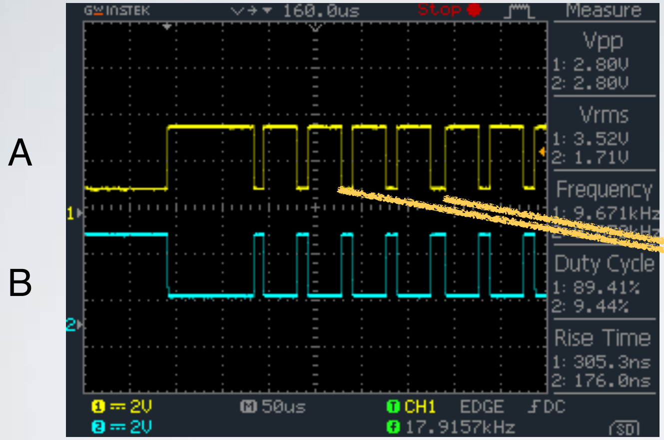
Can be idle between any slots

A set of up to 512 slots is assembled to form a frame

Each frame is prefixed by a control "control slot" with a start code

Start of the complete frame must be synchronised with receivers

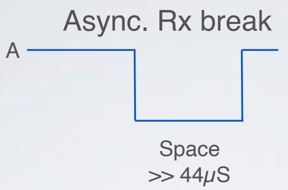
MARK AFTER SLOT



Varying time between slots (see page 82

Minimum: 0 s Maximum: 1 s
The maximum possible Mark-between-slot time is one second, after which the signal is considered to have failed.

ASYNCHRONOUS BREAK



A break is a run of 0s that exceeds the size of one character

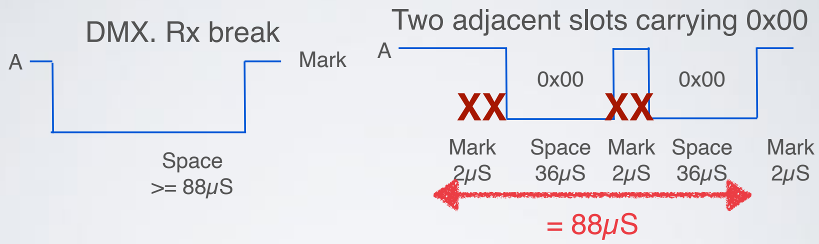
Breaks can be detected at the receiver (in UART Status Register)

Each break is followed by a ***Mark*** (by definition)

BREAK IN DMX512

Each frame starts with a ***DMX break*** (provides synchronisation)

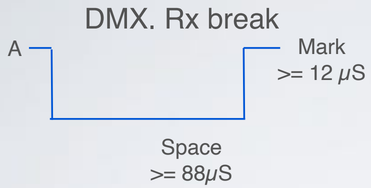
DMX defines a ***break at the receiver >*** **88 μS.**

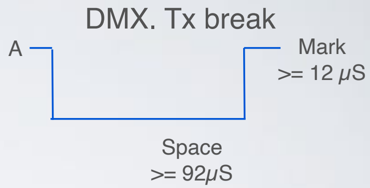


88 μS is longer than two 0x00 received slots with 4 errors.

At receiver, a received DMX break causes a UART "error"
A flag in the status register then indicates the start of a frame

BREAK SENT IN DMX512



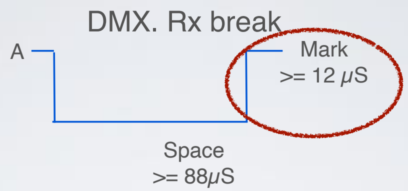


At the receiver, ***a break >*** 88 μS of continuous low indicates the start of a frame.

In DMX, ***the break at the sender*** is > 92 μS of continuous low

Why is the break duration specified as larger at the transmitter?

MARK AFTER BREAK



The break is followed by a 12 μS high level (Mark After Break)

The next low transition indicates the control slot

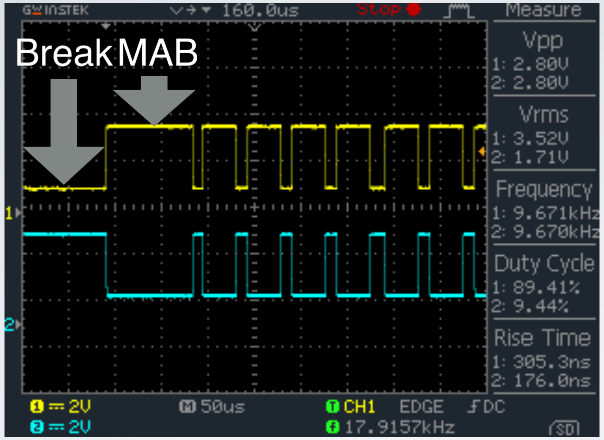
The control slot carries the **Start Code** value

The most common start code is 0 indicating "data"

MARK AFTER BREAK

A

B



See page 79 of Guidelines

The MAB allows time for slow receivers to process break

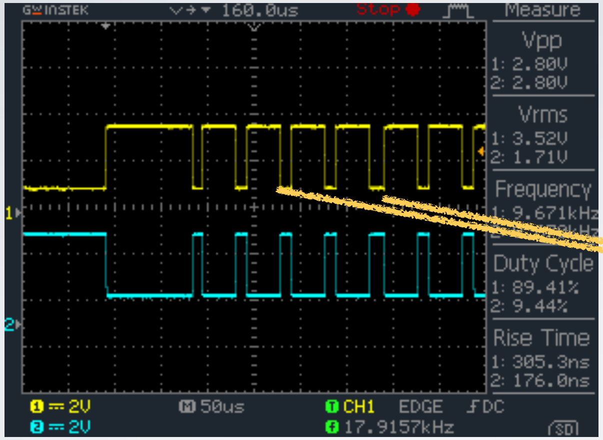
Minimum at sender: 12µs Minimum at receiver: 8µs

Question: What would happen if a receiver took more than 8µs?

MARK AFTER SLOT

A

B

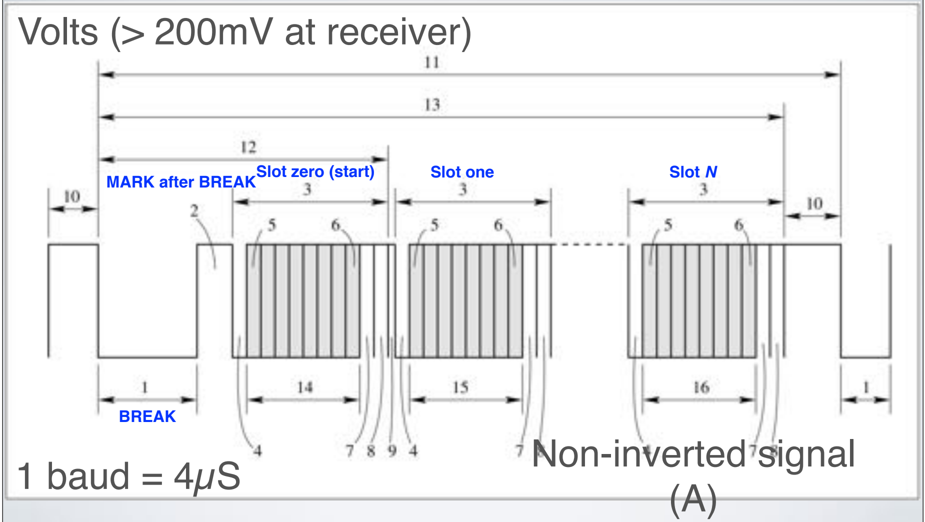


Varying time between slots (see page 82)

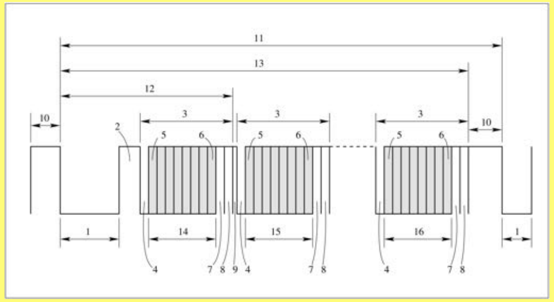
Minimum: 0 s Maximum: 1 s

The maximum possible Mark-between-slot time is one second, after which the signal is considered to have failed.

COMPLETE DMX

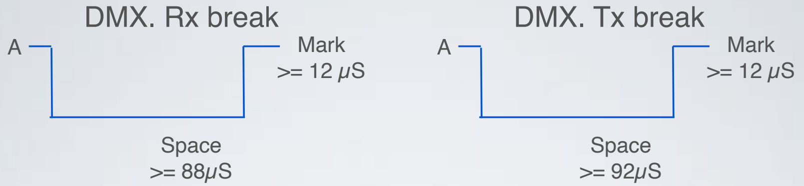


LEGEND



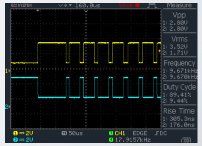
1. SPACE for BREAK
2. MARK after BREAK (MAB)
3. Slot Time
4. START bit
5. LEAST SIGNIFICANT Data BIT (LSB)
6. MOST SIGNIFICANT Data BIT (MSB)
7. STOP Bit
8. STOP bit
9. MARK time between slots
10. MARK before BREAK (MBB)
11. BREAK to BREAK time
12. RESET Sequence (BREAK, MAB, START Code)
13. DMX512 Packet
14. START CODE (SLOT 0 Data)
15. SLOT 1 Data
16. SLOT n DATA (Max. 512)

SUMMARY

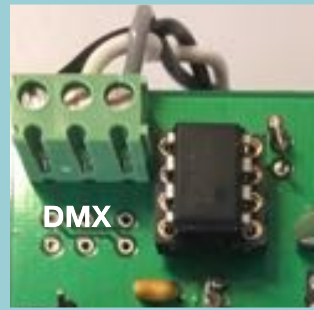


This has module has described:

- The asynchronous beak
- How DMX uses a break to indicate the start of frame
- How DMX chose the minimum specified DMX break duration
- The minimum mark after break
- The minimum/maximum time between slots



EIA-485 CONTROL BUS

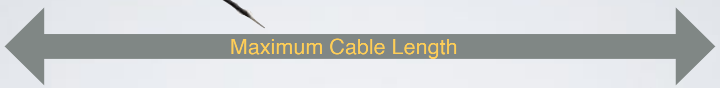


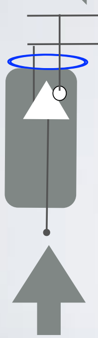
CABLE LENGTH

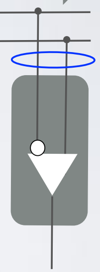
E1.27

Module 3.2

MAXIMUM CABLE LENGTH







Capacitance - limits maximum baud rate

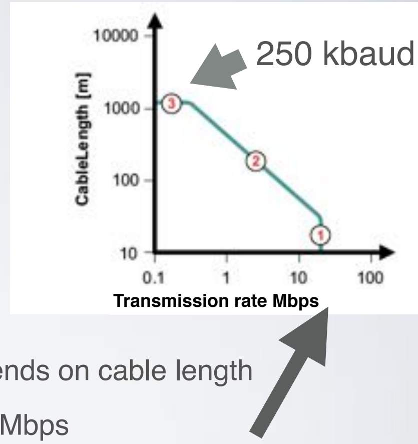
Attenuation/Metre - limits maximum length of cable

Losses - limits number of devices that can be connected (connector type and input impedance)

CABLE LENGTH

Signal strength, one receiver

= signal*attenuation/distance



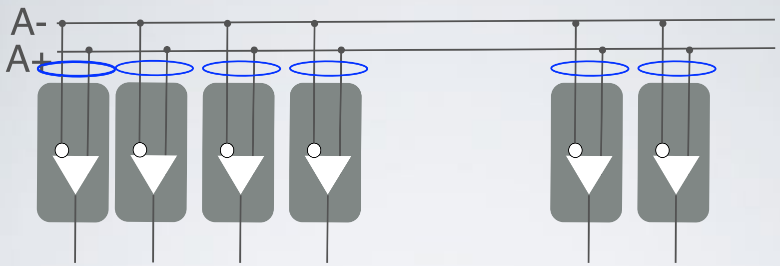
Max transmission baud rate depends on cable length

For very short distance ~10-40 Mbps

For moderate distances rate x length <10^7 (attenuation/metre)

For long distances, 250kbps, but cable attenuation dominates

LOAD FROM 32 RECEIVERS



Each standard EIA-485 rec has an input impedance of 12K.

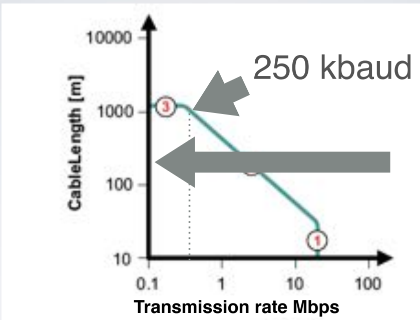
32 receivers placed in parallel present a combined load of 376 ohms.

Max load is a lot more than cable impedance!

RESISTANCE IN PARALLEL

- Basic reminder:
  - R in parallel with r =1/((R^{-1})+(r^{-1}))
  - Two resistances of resistance R in parallel = R/2
  - Four resistances of resistance R in parallel = R/4
  - Eight resistances of resistance R in parallel = R/8
  - .... 32 in parallel = R/32

32 RECEIVERS

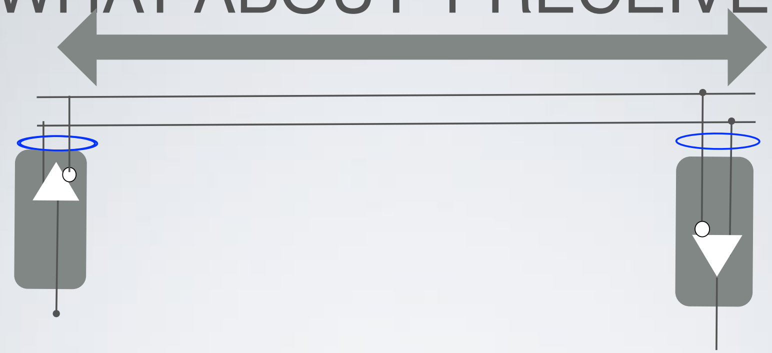


300m Safe distance with 32 receivers

~ 300m with 32 “standard” receivers (each Rx incurs signal loss)

32 receivers at 250 kbps limits bus to 300m

WHAT ABOUT 1 RECEIVER?



DOES 250 KBAUD WORK AT 1KM?

Start by looking at signal at transmitter

Power Margin (with no cable loss)
= 10 $Log_{10}[((V_{tx})^2/(V_{rx})^2]$ dB
= 10 $Log_{10}[(5x5)/(0.02 \times 0.02)]$ dB
= 38 dB

i.e. the signal is 38 dB above the receiver threshold

Now look at signal at transmitter

Actual signal at the receiver is reduced because of:
Cable attenuation, Loss at connectors, etc
We need a positive margin to take care of noise, and interference

DOES 250 KBAUD WORK AT 1KM?

1. Consider 300m & standard gauge conductors with 32 receivers @4dB/100m:
Propagation loss @ 300m = ~12dB
Receiver loss ~0.3dB (Total loss for 32 receivers = 10.4 dB
Total loss = 22.4 dB
Signal margin at receiver = 38 -22.4 dB = 15.6 dB
Positive margin sufficient to operate with noise/interference

2. Consider now 1000m & standard gauge conductors *@4dB/100m:*
Propagation loss @ 1000m = 40dB
Receiver loss ~0.3dB (Total loss for 32 receivers = 10.4 dB
Total loss = 50.4 dB
Signal margin at receiver = 38 -50.4 dB = - 12.4 dB
Negative margin - insufficient to reliably work !!!

Let's look at what we can change…

DOES 250 KBAUD WORK AT 1KM?

3. Consider larger gauge conductor (lower resistance/m)
@3dB/100m (depends on cable choice)
Propagation loss @ 1000m = 30 dB
Receiver loss ~0.3dB (Total loss for 32 receivers = 10.4 dB
Total loss = 40.4 B
Signal margin at receiver = 38 -40.4 dB = -2.4dB
Negative margin - insufficient to reliably work !!!

4. What about if we only had one receiver and low loss cable?
@3dB/100m (same as above)
Propagation loss @ 1000m = 30 dB
Receiver loss ~0.3dB (Total for 1 receiver = 0.3 dB
Total loss = 30.3 B
Signal margin at receiver = 38 -30.3 dB = 7.7 dB
Positive margin sufficient to operate with noise/interference

DMX can work over 1000m if using low loss cable and 1 receiver

EFFECT OF ERRORS

What happens if bauds become corrupted?

If any frame has detected errors ***the entire frame is ignored***

Some data errors could go un-noticed

A receiver might think everything is OK if slot data is corrupted

Each frame repeats all data slots values again in the next frame

Does it ***really*** matter if one frame is missed?

DMX MUST NOT be used for mission-critical applications

e.g. do not use for pyrotechnics or where lives might be at risk!

EIA-485 CONTROL BUS

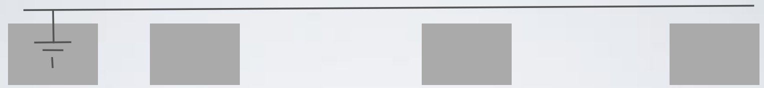
CABLE GROUND/SHIELD

E1.27

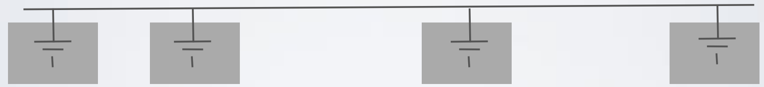
Module 3.2

GROUND/SHIELD

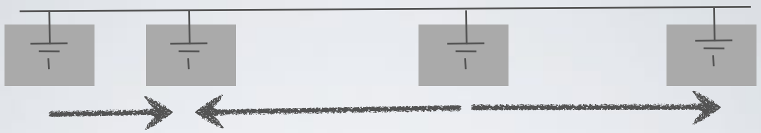
Required grounding of the shield



Why is this a bad idea?



GROUND LOOPS



Grounding the cable at each equipment causes problems:

- Equipment might (will) have a different ground potential
- A current will flow along the cable, and that disrupts communications

Only the output line at transmitter (controller) is grounded.

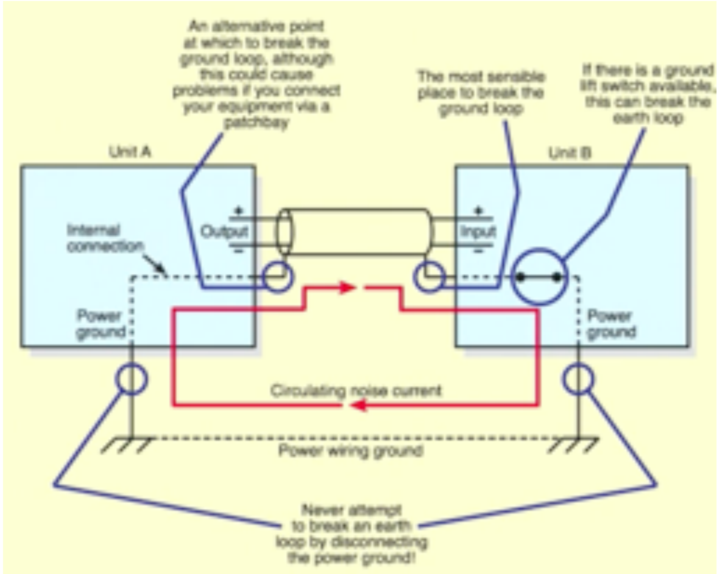
- The connectors must ***not*** be grounded at the receiver.
- The ***Shield*** does need to be connected through in and out connectors.

Earth Rod

AVOIDING GROUND LOOPS

Everything went well - until the whole rig went to full, and the DMX cable vaporised!

Recap: Grounding Equipment



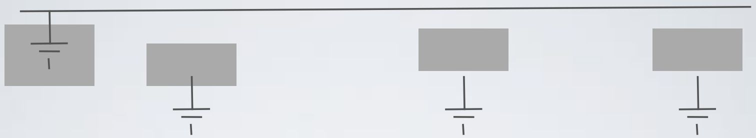
EIA-485 CONTROL BUS

RECEIVER DESIGN

E1.27

Module 3.2

AVOIDING GROUND LOOPS



Balanced lines do not connect the ***chassis grounds*** of different equipment:

Each receiver has TWO ground levels:

1) Local earth for electrical safety.

2) The communications bus shield

Each receiver ***decouple*** the transceiver through an ***opto-isolator***.

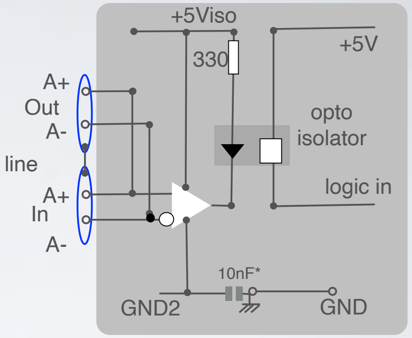
With isolation, the transceiver needs a separate the power supply

All comms circuitry is connected to one earth (at sender)

If no sender is driving the bus, the line floats to the level of a transient *

Transients can be many kV, so care is needed in this design.

REAL-WORLD RECEIVER



Receiver interface from line driver

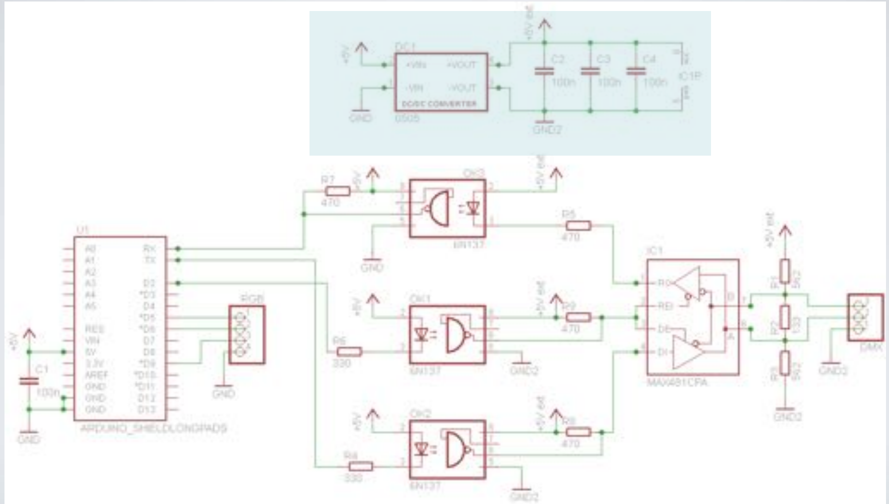
Opto-isolated transceiver

DC-DC conversion to isolate transceiver (Chasis, +5Viso)

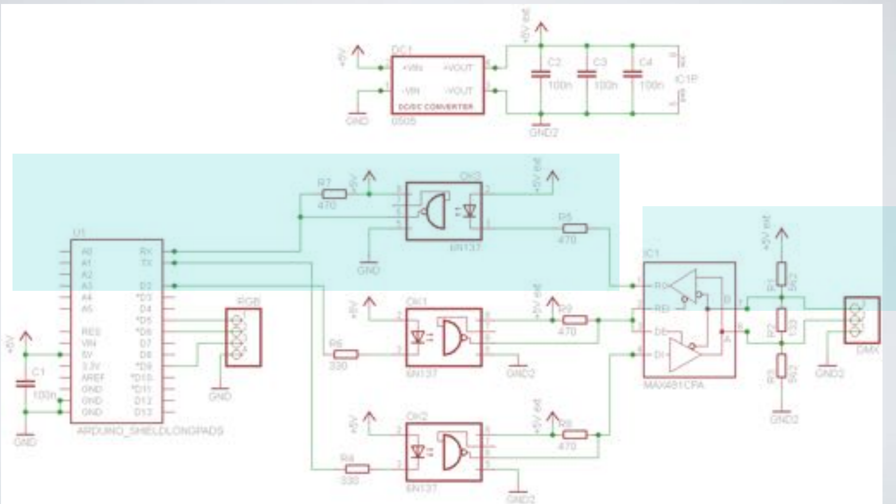
Could add capacitive coupling to ground 10 nF, 1kV (protects from faults)

*See DMX Guidlines, p39

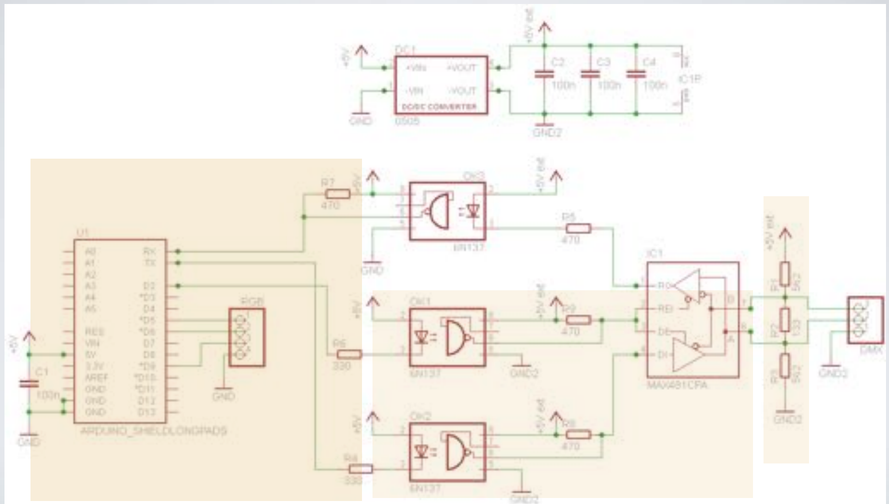
ISOLATED TRANSCEIVER



ISOLATED RECEPTION



ISOLATED TRANSMISSON



CONNECTION TO CABLE

Male Female

5-pin XLR connector

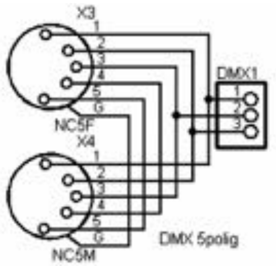
Pin 1 (screen/GND)--------------------- Pin 1 (screen/GND)
Pin 2 (A -)---------------------------------- Pin 2 (A -)
Pin 3 (A +)---------------------------------- Pin 3 (A +)
Pins 4,5 ----------------------------------- Usually not used

Equipment has male and female sockets

Electrical connection between in and out

This design fails safe if you turn off equipment!

The bus has a wire connection end-to-end



Equipment Connection

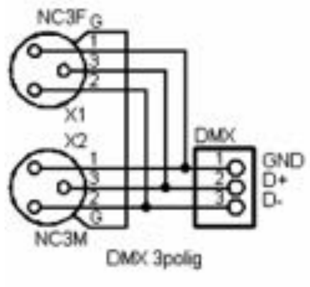
CONNECTION TO CABLE

3 XLR connector

Pin 1 (screen/GND)--------------------- Pin 1 (screen/GND)
Pin 2 (A -)---------------------------------- Pin 2 (A -)
Pin 3 (A +)---------------------------------- Pin 3 (A +)

Both 3 and 5 pin versions are popular

(we'll use both in the labs)

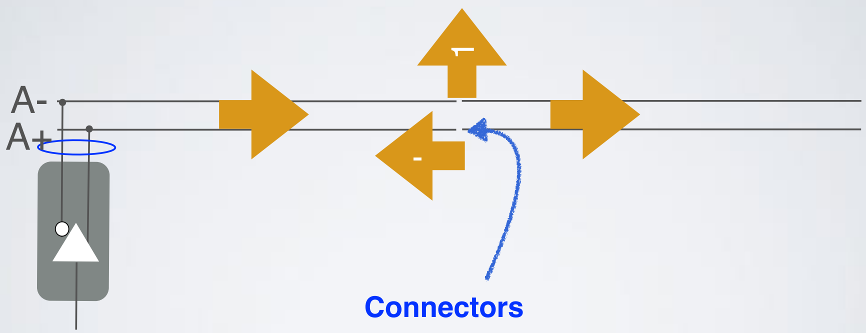


BUS TERMINATION

Module 4.2

SIGNAL PROPAGATION

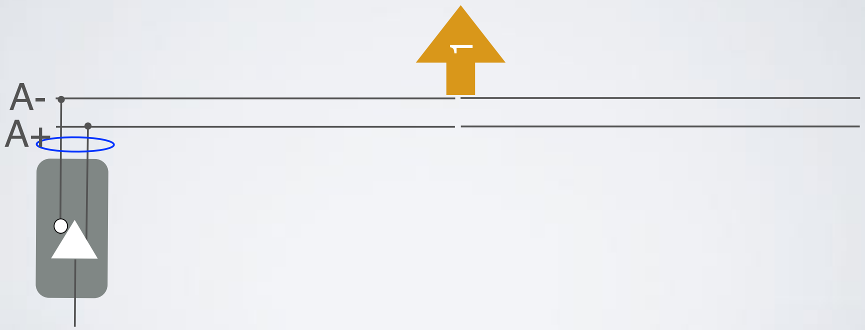
(1) What happens to the signal when we join two cables?



SIGNAL PROPAGATION

(1) What happens when we join two cables? - loss

(2) What happens as the signal travels along the cable?

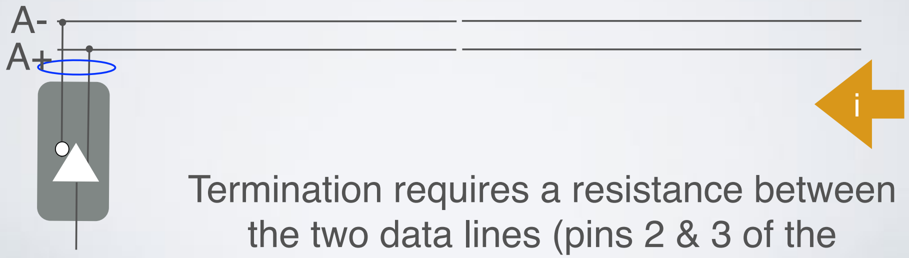


SIGNAL PROPAGATION

(1) What happens when we join two cables? - loss

(2) What happens as the signal travels along the cable?

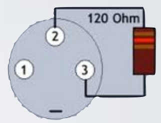
(3) What happens to the signal at the end of the cable?



Termination requires a resistance between the two data lines (pins 2 & 3 of the connector)

TERMINATION

The termination impedance value should match the cable characteristic Impedance.



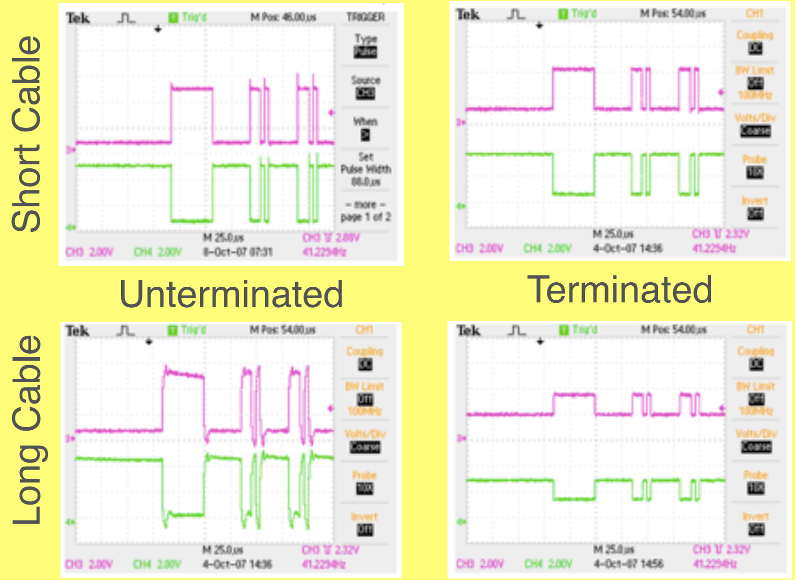
Termination of the cable with the characteristic impedance causes no reflections of the transmitted signal.

When the cable is cut to ***any length*** and ***terminated***, measurements will be identical to values obtained from an infinite length cable.

The resistor should be rated at least 0.2W.

See guidelines p22

EFFECTS OF TERMINATOR



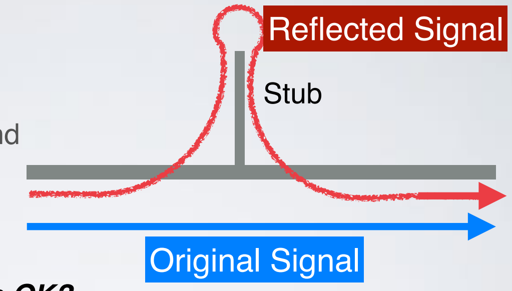
EFFECT OF CABLE STUB

Signal split at stub

Half the signal energy travels along the stub

Reflected at stub end and travels back down stub

Reflection propagates with original signal



How long a stub can be OK?

Assume that the reflected signal needs to not be more than 10% delayed relative to the original and that

v= 0.6-0.8 and c= 3 x10^8 m/s

Lstub ≤ (Tr/10) x v x c

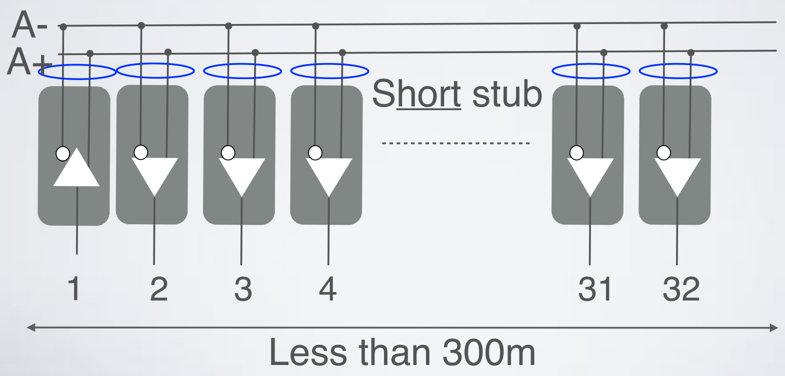
Lstub for 250ns Tr (@250kbaud) = 6m

Lstub (@1Mbaud) = 2m

Most buses have several stubs, best to keep all ***SHORT***

EIA-485 CABLES

Up to 32 receivers can attach directly to cable



EIA-485 SIMPLEX EQUIPMENT BUS: DMX-512 FRAMES

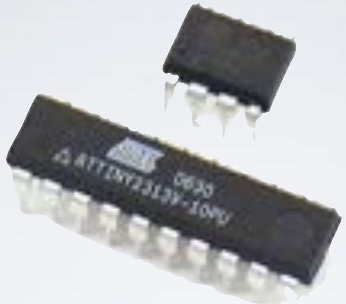
Module 5.0

DMX RECEIVER HARDWARE

Module 5.3

ATMEL* AVR (1997)

8-BIT MICROCONTROLLER



A complete computer on a chip with serial communications

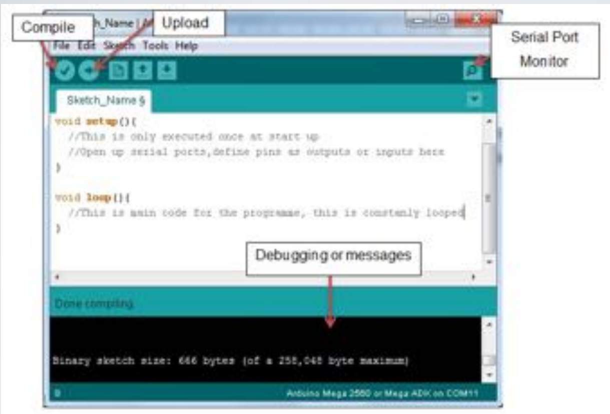
Named after Alf (Egil Bogen) and Vegard (Wollan)

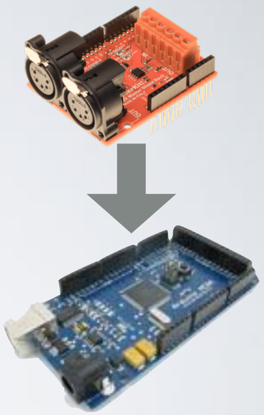
2003: 500 Million sold in first 5 years

2005: Arduino appeared, over 700,000 sold

*ATMEL is now MicroChip

ARDUINO DMX





Total cost about £15-£30, free development tools!

DMX Shields cost ~£20

AT MEGA 8515-16

AMTEL AVR Core

2.7 - 5.5 Volt, 16 MHz (16 MIPS)

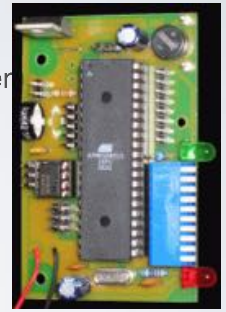
130 instruction RISC processor, 32 register

8 KB program Flash Program Memory

512 B internal SRAM, 512 Byte EEPROM

35 general purpose I/O lines

Serial Programmable USART



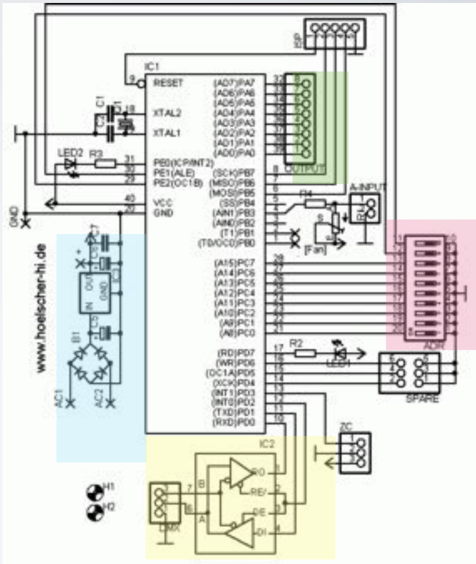
http://www.atmel.com/

Cost about £2-£3, free development tools!

SIMPLE DMX RECEIVER

All with minimal logic!

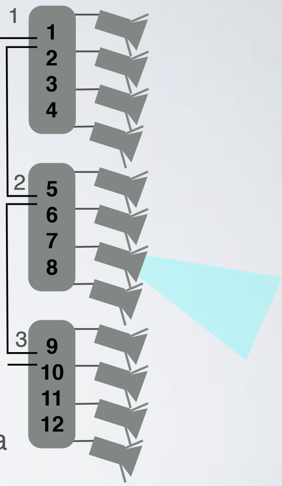
- AVR 8515 Microcontroller
- 5V Power Supply
- EIA-485 Driver/Receiver
- Parallel input (DIP switch)
- PWM/Level output



DMX ADDRESSING AND RECEIVERS

Module 5.2

LOGICAL BUS



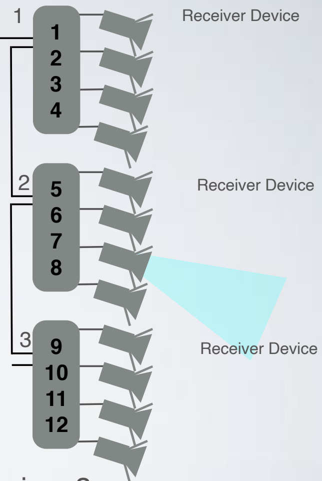
All devices controlled by one transmitter

Each device receives the same DMX data

Each device uses only a portion of the DMX data to control its own output(s)

ADDRESSING A RECEIVER

Control Surface



Each receiver allocated a base address

In this case, the receiver uses 4 slots.

e.g. here, Receiver 2 uses address 5

Hence Slots 5-7 of the frame used by Receiver 2

This receiver uses these 4 slots to control its four outputs

DMX SLOT ADDRESSING

2^0	2^1	2^2	2^3	2^4	2^5	2^6	2^7	2^8

least significant bit first

DMX addresses are often setup using DIP switches:

- Switch setting 100000000, = 1
- Switch setting 101000000, = 5
- Switch setting 111000000, =7

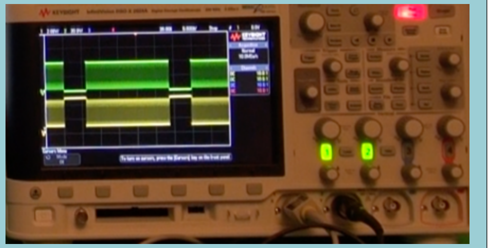
Checks these switch settings for yourself:

A DMX base address of 40 sets 4,6

A DMX base address of 393 sets 1,4,8,9

FRAMES OF SLOTS

Control Slot
Slot 1
Slot 2
Slot 3
Slot 4
....
Slot 512



Module 5.1.2
Demo Measuring the Frame Rate

MAXIMUM FRAME RATE

Total frame duration = Break+Mark_after_break+slot*(n+1)

= 92 + 12 + (44 * 513) μS

= 22 676 μS (for full 512 B frame)

Maximum frame rate = 44 frames /sec

Lower rates common for actual operation

e.g. 15 or 30 frame/sec

Allows time between slots

Maximum information transfer rate = 512 x 30 (30 frame/sec)

122.88 kbps (i.e. data bits/second)

SMALLER-SIZED FRAMES

Many applications send 512 B frames, but frames can be smaller.

The receiver knows it has reached the end of frame when it sees the break marking the start of the ***next*** frame.

A smaller frame size allows a higher rate

Small frames are also used for certain types of control slots.

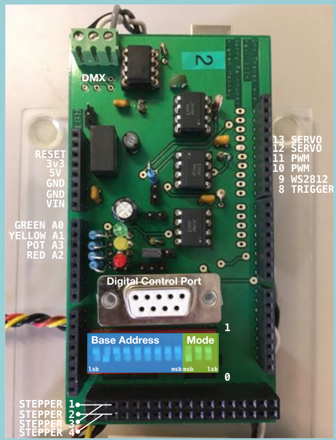
MULTIPLE CHOICE

1) Which of the following is true for DMX?
(a) DMX uses bidirectional transmission
(b) Asynchronous communication sends 3 extra overhead bauds per byte
(c) A sender can pause between each asynchronously sent byte
(d) The stop baud is the same level as for an idle cable

2) Which of these is true for DMX cables?
(a) The cable uses a pair of conductors to send the signal
(b) The cable must be shielded
(c) The cable must be earthed at ***every device*** connected to the bus
(d) The bus must be terminated at ***both ends of the cable***

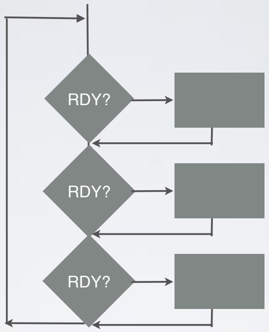
3) Which if these is true of the 120 Ohm EIA-485 bus?
(a) A typical input impedance for a transceiver is 12k Ohms
(b) The maximum number of receivers is determined ***only*** by the cable length
(c) A longer length of cable will deliver acceptable performance with fewer

DMX RECEIVER SOFTWARE



Module 5.4

POLLING

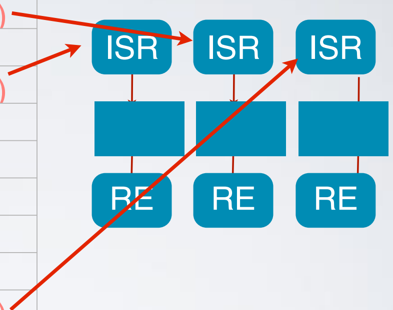


Polling

- Difficulty in responding quickly to input
- Tricky when something more important, longer, etc

INTERRUPT VECTORS

Vector	Location	Value
Reset	$000	
Ext Int 0	$001	(ISR2)
Ext Int 1	$002	
Timer 1	$003	(ISR1)
T1 cmp A	$004	
T1 cmp B	$005	
T1 Oflow	$006	
T0 Oflow	$007	
SPI done	$008	
USART	$008	(ISR3)



Initialise a set of vectors to point to ISRs

Write start address of each routine into corresponding locations

SOFTWARE DESIGN

- System functions:

  Initialise hardware - sets I/O pins, clock, USART, Timer, etc.

  Initialise software - setup vectors, initialise data

  Monitor user interface

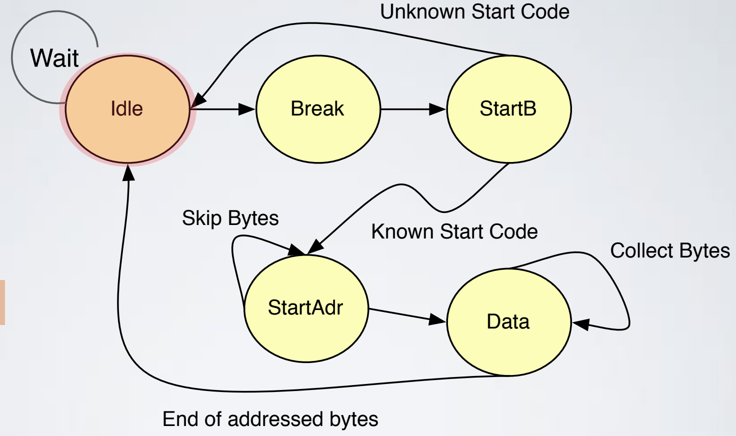
  Output Status display

  Receive DMX Signal

  Output Control waveform

  Check program is running (watchdog)

DMX RECEIVER STATES



States

Idle

Break

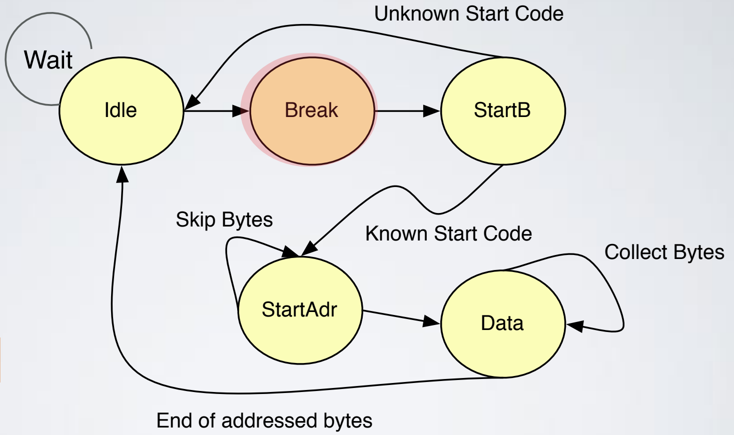
StartB

StartAdr

Data

Note: All errors also transition to the Idle state

DMX RECEIVER STATES



States

Idle

Break

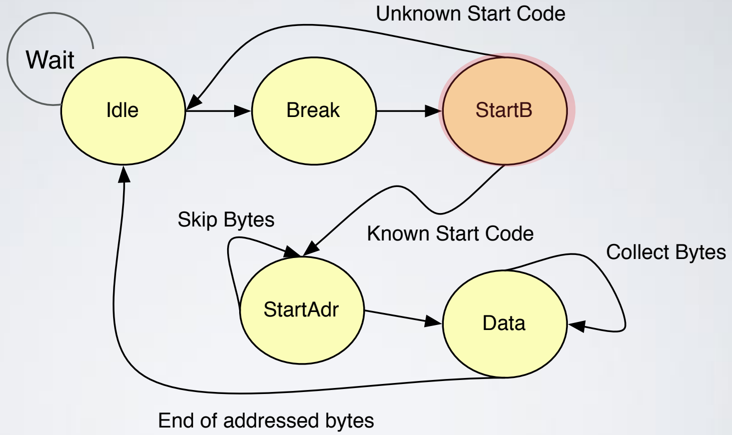
StartB

StartAdr

Data

Note: All errors also transition to the Idle state

DMX RECEIVER STATES



States

Idle

Break

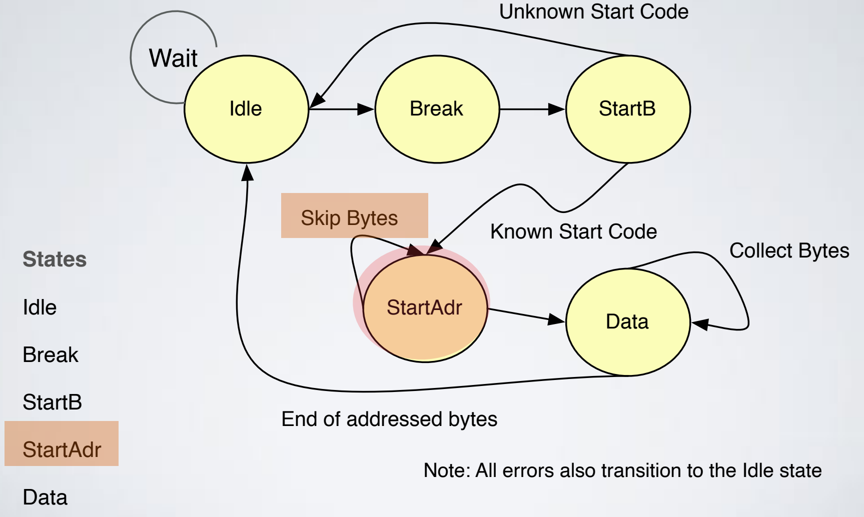
StartB

StartAdr

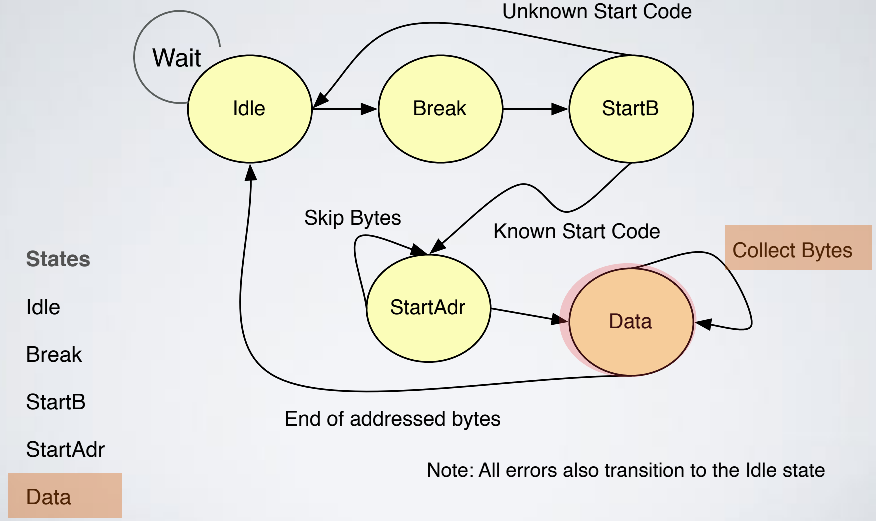
Data

Note: All errors also transition to the Idle state

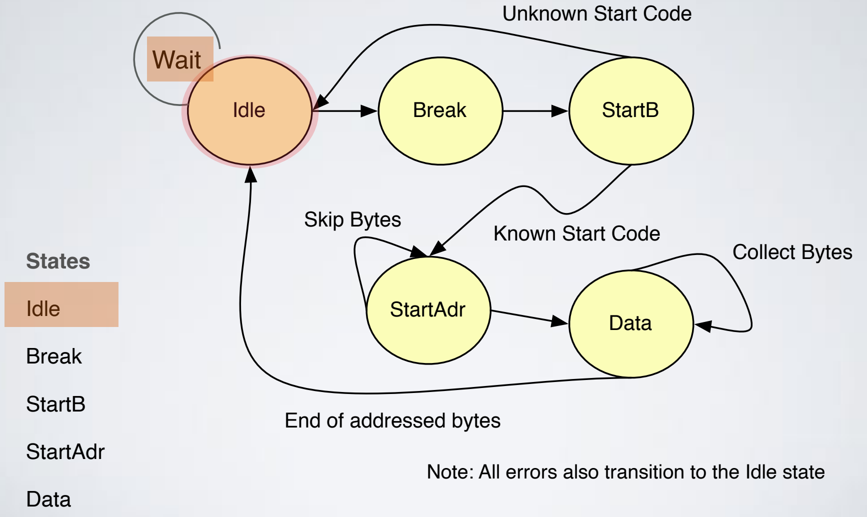
DMX RECEIVER STATES



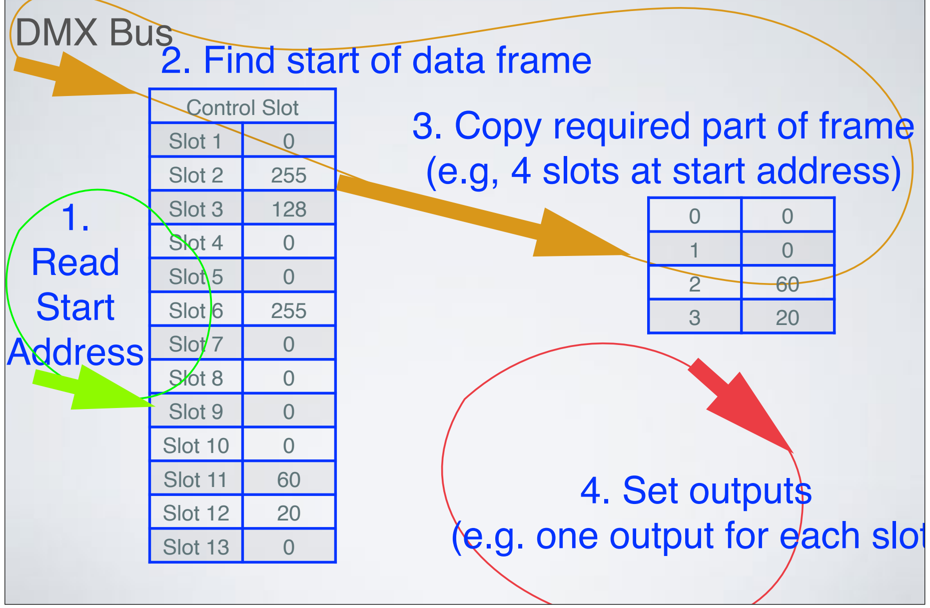
DMX RECEIVER STATES



DMX RECEIVER STATES



DMX - EXAMPLE RX CODE



Control Slot	
Slot 1	0
Slot 2	255
Slot 3	128
Slot 4	0
Slot 5	0
Slot 6	255
Slot 7	0
Slot 8	0
Slot 9	0
Slot 10	0
Slot 11	60
Slot 12	20
Slot 13	0

0	0
1	0
2	60
3	20

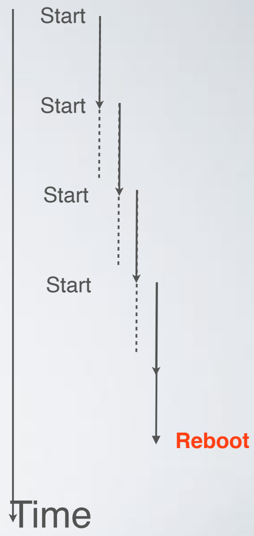
WATCHDOG TIMER

A simple timer that when triggered, counts down to zero and triggers a reset interrupt:

Timer initialised at start.

Periodically reset & restarted by main program.

If the timer ever reaches zero, the program is assumed to have crashed and the watchdog Interrupt service triggers a full reboot.



DMX MAIN VARIABLES

Hardware registers:

int UCSRA // The Status Register of the UART

char DMXByte

Variables Used:

int DMXAdress // Read from the DIP Switch

int DmxState: {Idle,Break,StartB, StartAdr, Data}

char Array DMXRxField[4]

int DmxCount // Used as a counter

DMXRxField:

0	0
1	0
2	60
3	20

DMX RECEIVER

This routine handles reception of DMX frames from USART.

Requires a state machine (*DmxState*) to know which parts of the frame have already been received.

This is a fairly “classic” communications protocol design.

Updates *DmxRxField[]* based on contents of DMX Frame.

It could be made more sophisticated by checking the timing constraints for reception of the data slots.

DMX RECEIVER

- ISR (UART_RX_vect)

```
{
static  uint16_t DmxCount;
   uint8_t  USARTstate= UCSRA;              //get state before data!
   uint8_t  DmxByte   = UDR;            //get data
   uint8_t  DmxState  = gDmxState;

if (USARTstate &(1<<FE))                        //check for break
   {
   UCSRA &= ~(1<<FE);          //reset flag
   DmxCount =  DmxAddress;          //reset channel counter
                               //(count channels before start address)
   gDmxState= BREAK;
   }

else if (DmxState == BREAK)
   {
   if (DmxByte == 0) gDmxState= STARTB; //normal start code detected
   else          gDmxState= IDLE;
   }
```

DMX RECEIVER

```
else if (DmxState == STARTB)
  {
  if (--DmxCount == 0)          //start address reached?
    {
    DmxCount= 1;                  //set up counter for required channels
    DmxRxField[0]= DmxByte; //get 1st DMX channel of device
    gDmxState= STARTADR;
    }
  }

else if (DmxState == STARTADR)
  {
  DmxRxField[DmxCount++]= DmxByte;          //get channel
  if (DmxCount >= sizeof(DmxRxField))    //all ch received?
    {
    gDmxState= IDLE;                //wait for next break
    }
  }
Retl
}
```

RECEIVER ISR ALGORITHM

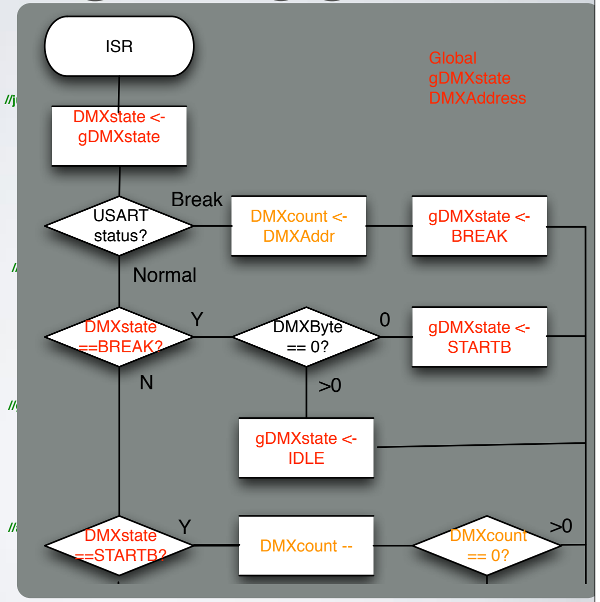
```
    ISR (UART_RX_vect)
{
static  uint16_t DmxCount;
    uint8_t  USARTstate= UCSRA;
    uint8_t  DmxByte   = UDR;
    uint8_t  DmxState  = gDmxState;          //j

if (USARTstate &(1<<FE))
   {
   UCSRA &= ~(1<<FE);
   DmxCount =  DmxAddress;
   gDmxState= BREAK;
   }

else if (DmxState == BREAK)
   {
   if (DmxByte == 0) gDmxState= STARTB;          //
   else              gDmxState= IDLE;
   }

else if (DmxState == STARTB)
   {
   if (--DmxCount == 0)
     {
     DmxCount= 1;
     DmxRxField[0]= DmxByte;          //
     gDmxState= STARTADR;
     }
   }

else if (DmxState == STARTADR)
   {
   DmxRxField[DmxCount++]= DmxByte;
   if (DmxCount >= sizeof(DmxRxField))          //
     {
     gDmxState= IDLE;
     }
   }
}
```



RECEIVER ISR ALGORITHM

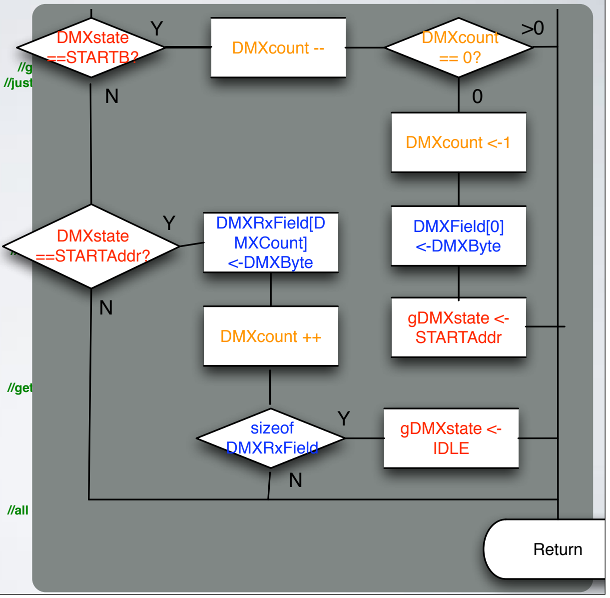

```
    ISR (UART_RX_vect)
{
static  uint16_t DmxCount;
    uint8_t  USARTstate= UCSRA;
    uint8_t  DmxByte   = UDR;                //g
    uint8_t  DmxState  = gDmxState;          //just

if (USARTstate &(1<<FE))
   {
   UCSRA &= ~(1<<FE);
   DmxCount =  DmxAddress;
   gDmxState= BREAK;
   }

else if (DmxState == BREAK)
   {
   if (DmxByte == 0) gDmxState= STARTB;
   else              gDmxState= IDLE;
   }

else if (DmxState == STARTB)
   {
   if (--DmxCount == 0)
     {
     DmxCount= 1;
     DmxRxField[0]= DmxByte;          //get
     gDmxState= STARTADR;
     }
   }

else if (DmxState == STARTADR)
   {
   DmxRxField[DmxCount++]= DmxByte;
   if (DmxCount >= sizeof(DmxRxField))          //all
     {
     gDmxState= IDLE;
     }
   }
}
```

DMX MAIN ROUTINE

Initialise hardware - sets I/O pins, clock, USART, Timer, etc.
Initialise software - zero **DmxRxField** array
Setup ISR for UART to load **DmxRxField** array
Setup ISR for output to use **DmxRxField** array
Enable watchdog
Enable Interrupts

Loop:
- Maintain user interface (switches, LEDs, etc)
- Sleep;
- Reset ***watchdog timer***
- *Goto Loop*

This program loops continuously, once initialised.

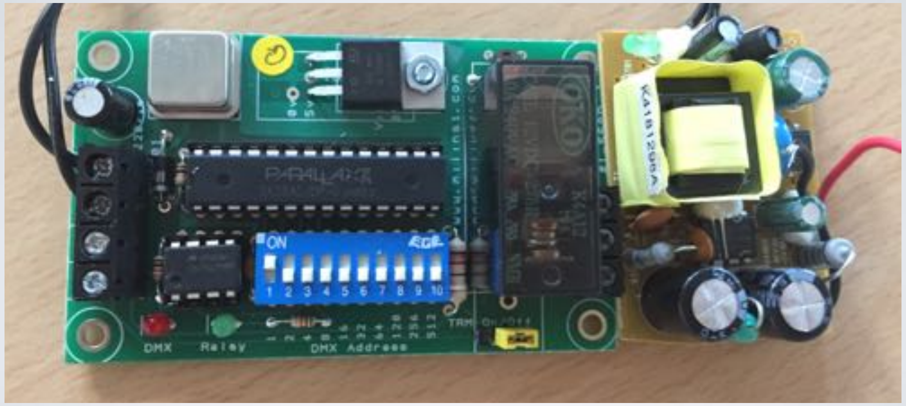
One ISR implements a watchdog timer to restart after a crash

DIGITAL OUTPUTS & RELAY CONTROL

Control Slot
Slot 1
Slot 2
Slot 3
Slot 4
....
Slot 512

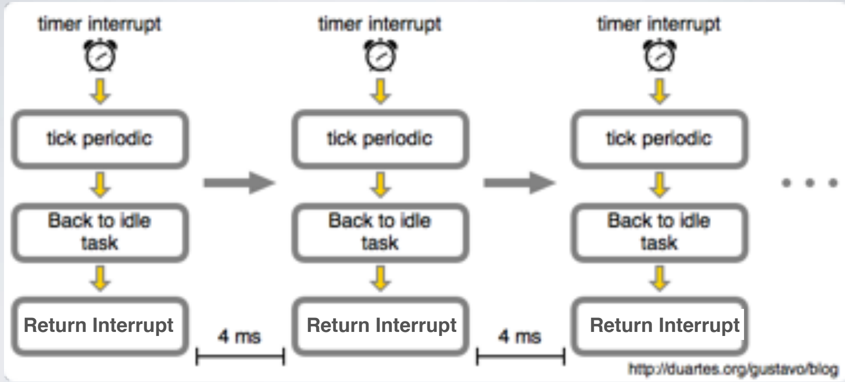
Module 5.5

SWITCHED RELAY



Bus Receiver   Microcontroller   Relay   Power Supply

OUTPUT A WAVEFORM



Each clock tick outputs next value
4ms = 250 Hz; 10ms = 100 Hz; 20ms = 50 Hz

EIA-485 SIMPLEX EQUIPMENT BUS: DMX512 CONTROL

Module 6.0

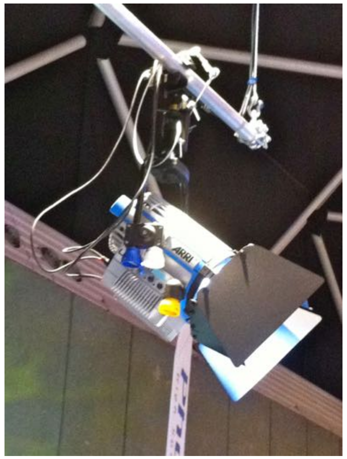
CONTROLLING POWER

Module 6.1

LIGHTING LEVEL

1 lux = 1 lumen per m^2

Moonless Night 0.004 Lux
Full Moon, clear night 1 Lux
Living Room 50 Lux
Office Lighting 500 Lux
Stage > 500 Lux
Overcast Day, 1,000 Lux
Spotlight 2,000 Lux
Dull Daylight 10,000 Lux
Direct Sunlight 100,000 Lux



WHAT ABOUT MAINS?

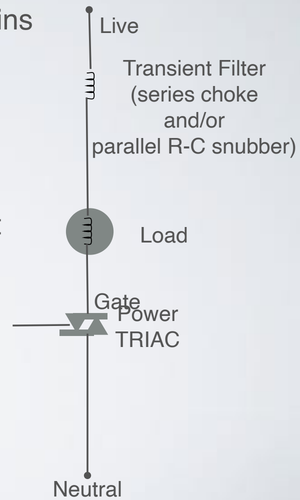
Traditional lamps driven directly from the mains

Need more control than on/off

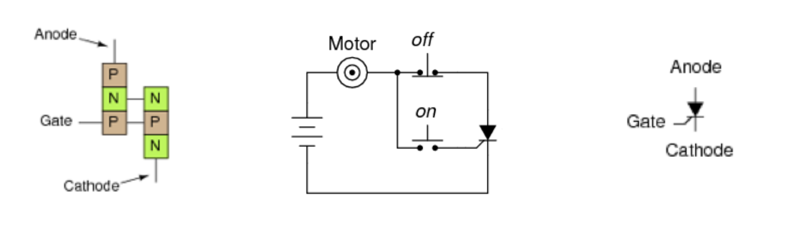
Normal method uses a TRIAC dimmer circuit

Gate fires Triac, turning on load

Choke/RC-Snubber suppresses interference



SILICON CONTROLLED RECTIFIER (SCR)



SCR fires when gate voltage is above a ***threshold***

Current flows from Anode to Cathode

This turns on load

Conduction continues until current ceases to flow ($I_{fwd} > I_H$)

The device functions as a ***latch***

AC TRIAC (THYRISTOR)

A TRIAC is effectively two SCRs

- allows AC operation

For high power, important:

- TRIAC fires ***cleanly***
- Turns-off at end of cycle

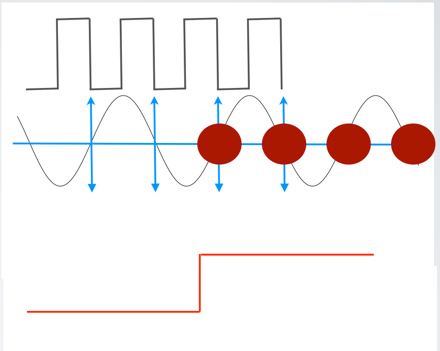
TRIAC equivalent circuit *TRIAC schematic symbol*

MAINS TRIAC SWITCH

Zero-Crossing Sync

Mains Cycle

+5V Trigger to Gate



Switching is at zero-crossing point (no current flowing)

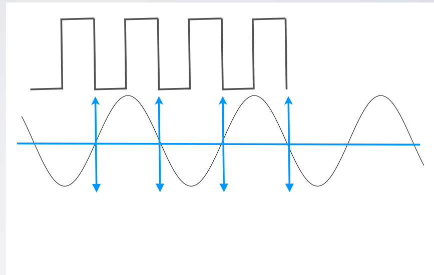
TRIAC "fired" after each zero-crossing when enabled (red)

TRIAC always switches off at end of each half-cycle

MAINS DIMMER - ZERO X

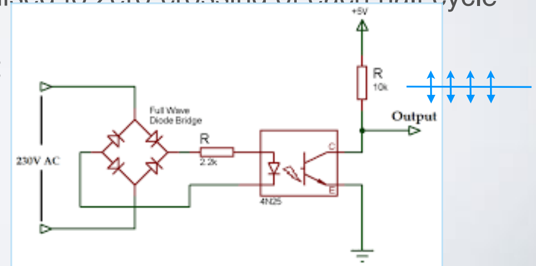
Zero-Crossing Sync

Mains Cycle



Switch-on is synchronised to zero-crossing of each half cycle

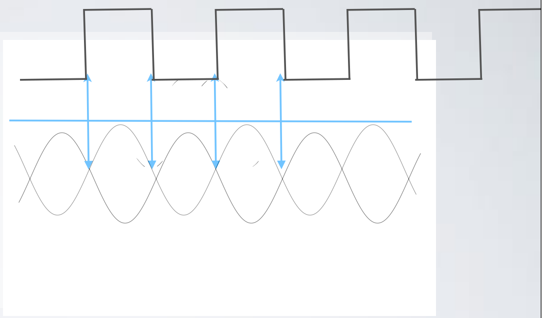
Example. simple circuit



2.MAINS DIMMER - ZERO X

Zero-Crossing Sync

Mains Cycle



Example. using +ve and -ve transformer outputs

Switch-on is synchronised to zero-crossing of each half cycle

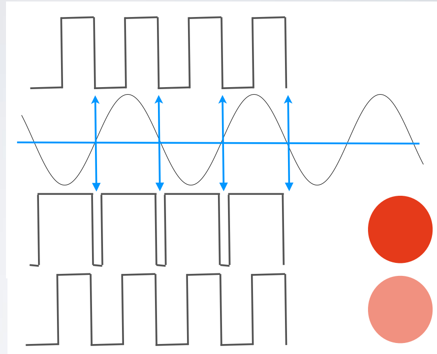
MAINS DIMMER

Zero-Crossing Sync

Mains Cycle

TRIAC Gate Trigger 95%

TRIAC Gate Trigger 50%



A mains “dimmer” works at 100Hz (50 Hz mains)

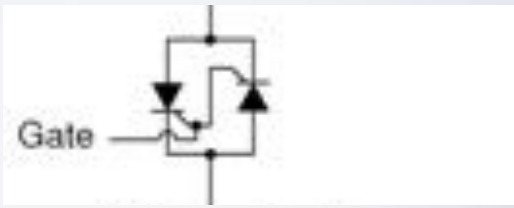
Gate Trigger is a 100 Hz PWM signal aligned to crossing point

Varies the start time of the pulse that fires the power TRIAC

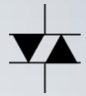
TRIGGERING THE TRIAC

The gate signal needs to be:

- Have a OV at the time of zero current
- Have an on voltage at the position in the mains cycle where the TRAIC is to fire
- The On-signal needs to rapidly force the TRIAC into conduction



DIAC

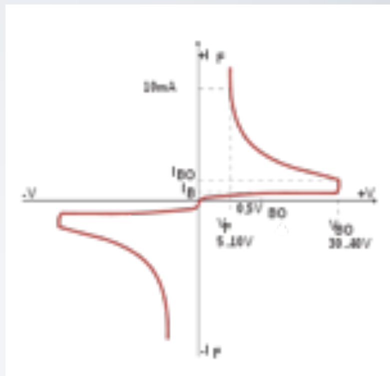


A DIAC resembles two diodes combined for AC operation

Conducts only above a threshold

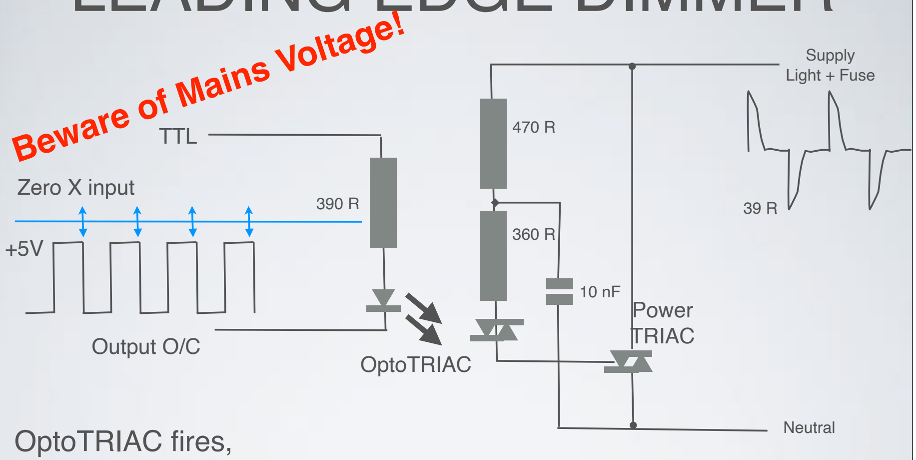
Opto-TRIACs are effectively a DIAC triggered by light level (from a LED)

Provides an easy way to reliably trigger a TRIAC gate.



This effectively operates as a threshold voltage trigger

LEADING EDGE DIMMER



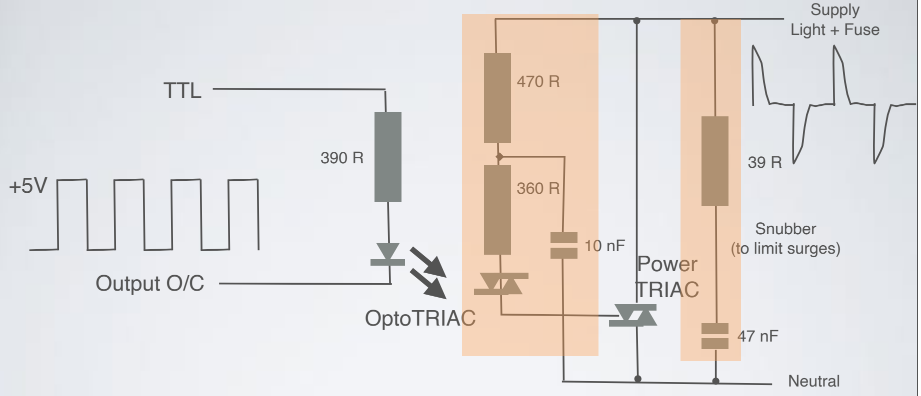
OptoTRIAC fires,

DIAC triggers at a threshold voltage

Dumping charge to the gate of the power TRIAC

TRIAC conducts once triggered, until supply returns to zero

RELIABLE TRIGGER

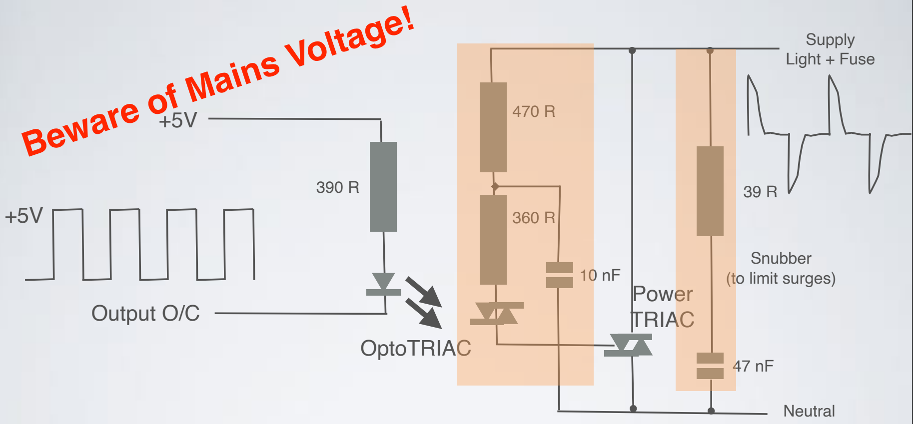


Snubber filters transients, preventing false triggers

10nF capacitor charges at start of cycle

Choke/Filter limits switching surges

RELIABLE TRIGGER



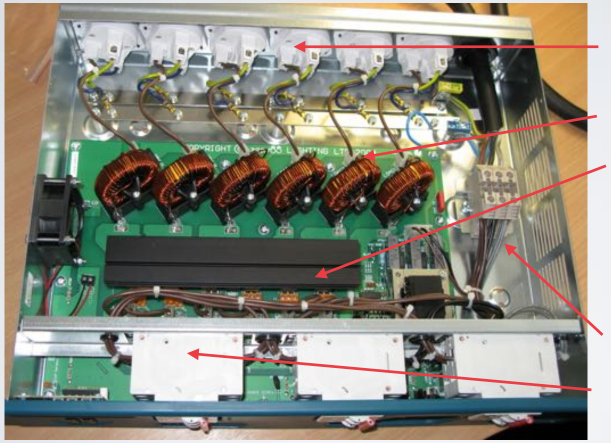
PROFESSIONAL DIMMER

Input: 3-Phase supply (3 x 32A)

Output: 6 Channel each at 10A (2 per phase)

Control: DMX (with RDM); CAN (ChilliNet)

PROFESSIONAL 6 CH



Output:

6 Channel at 10A

RF filter

Triac

Input:

2 Channel / phase

6 Circuit Breakers

Transfer Function -i.e. Dimmer Curve

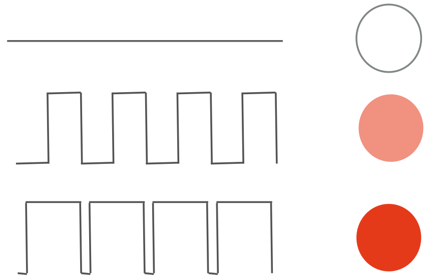
How does the microcontroller map a slot value to a fine signal for the TRIAC?

- Actually there are different possibilities: e..g one way:

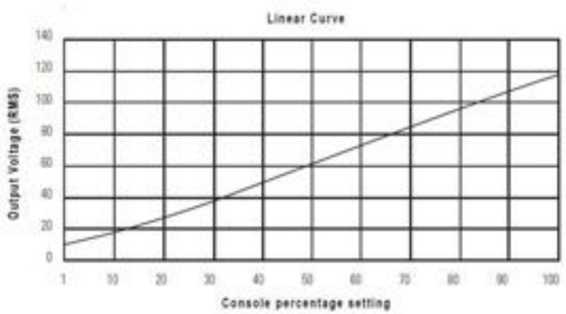
0x00
TRIAC Gate Trigger 0%

0x7F
TRIAC Gate Trigger 50%

0xF2
TRIAC Gate Trigger 95%



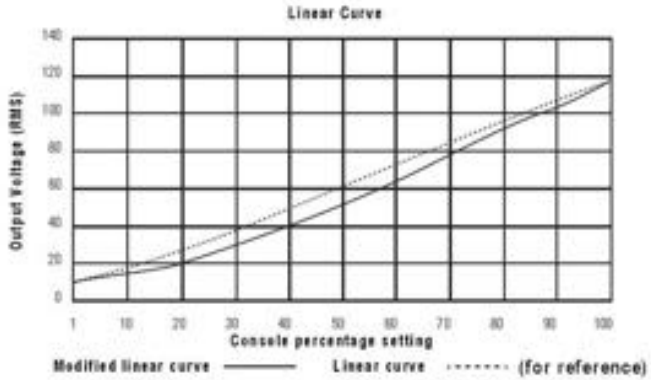
Linear



1:1 Ratio

Control input percentage to Root Mean Squared (RMS) voltage output

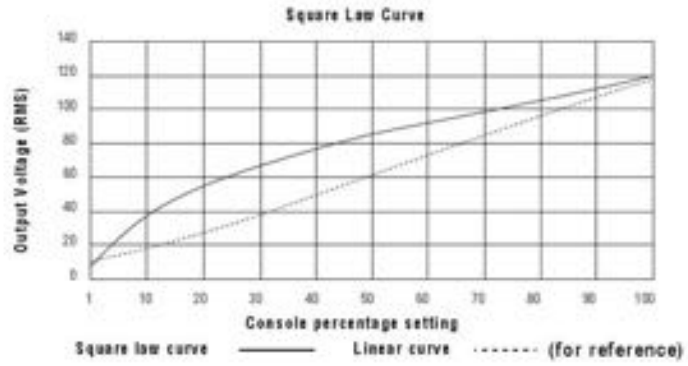
Modified Linear



Output does not have to be proportional to the control value.

Improved control at low levels for better performance in low-wattage fixtures.

Square law curve



Improved control at low values.

A square law curve applies a multiple derived from the square root of the control level (with full output equal to 1.00) to increase voltage response at low control levels to compensate for the infrared loss of an incandescent lamp.

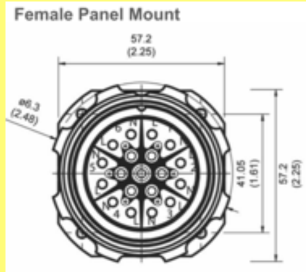
SOCAPEX CABLE

19 pin:

6 x 3 23A circuits

Each circuit L,N,E

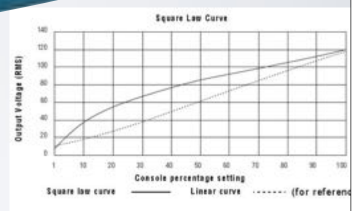
Circuit breaker / fuse for each output



SUMMARY

- **We talked about:**
  - SCR, TRIACs, DIACs, OptoTRIAC
  - Firing Triac, Zero-Crossing synch, Snubber and Filters
- **TRIAC Control**
  - TRIAC Dimming Output (“random” turn-on within cycle)
  - TRIAC Switching Output (“ZC” turn-on at start of

CHECK THE COURSE MODULE FOR DETAILS OF OPERATION



ELECTRICAL SPECS - IET WIRING REGULATIONS

IEC 60364 sets overarching rules. BSI is a member of CENELEC ... likely that most requirements are the same across Europe. Usually the base spec is adopted with a name change, usually just add EN.

RCD:
All sockets not exceeding 32A now needs RCD 30mA unless justified (for non household) applications.
Regulation requires now for new dwellings to have RCD for luminaires.

Arc fault detection now also recommended for wooden building etc. Surge arrestors to be installed to protect over voltage from HV switchover where effects public services and injury to life. These are placed in parallel with other equipment across the supply Not likely needed for outdoor events, but actually easily fitted.

Cable runs:
Run cables along the top of a truss are ok. There is now a recognised danger in case of trunking made of plastic that melts and cables fall then possibly entangling people escaping fire or firefighters. Use metal saddles to fix. Do not tape or tie around a bundle around a doorway - instead wrap around a scaffold pole or use a metal hook to hold.

Do not mix MCBs from different manufacturers - they may fail differently - not likely an issue on the first fault, but can degrade surrounding components - eg some MCBs venting to left, some to right - mixing them can be a concern that people need to sign off as a consideration. Each manufacturer will already have tested their own system.

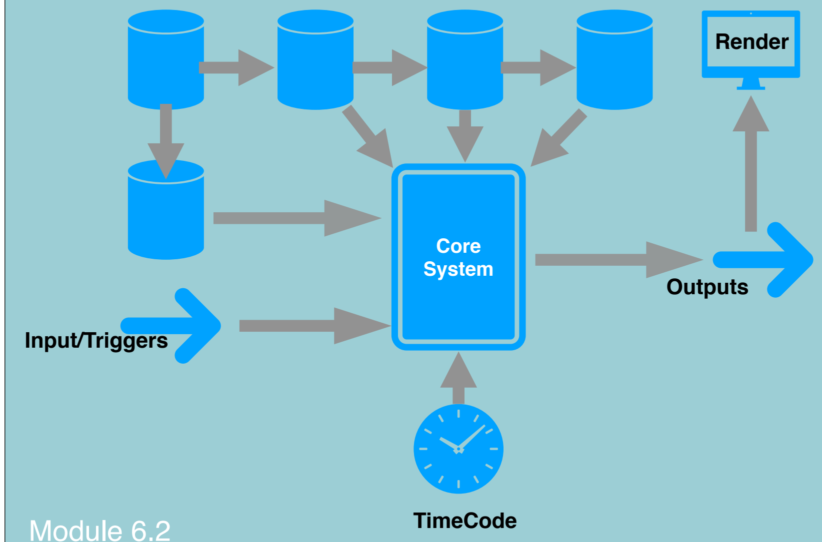
Section 444 - interference. Earthing arrangements - armouring on cables can have significant impedance. When used as the earth ... This can result in ground loops. Audio hum and worse for digital signals.

Challenges of switch-mode power supplies embedded in modern equipment - that present much more complex loads on the supply.
Noise also easily impacts triac firing.

Rubber power cables bs7909 update. Use rubber cables were possible, or avoid any mechanical damage. Armoured cables were needed. - note armouring impedance unless using a separate earth.

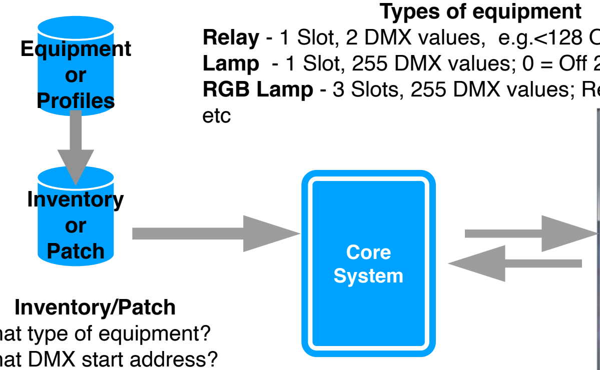
There are type a and b RCD that need to be considered. Not recommended to mix types —- in Germany this is forbidden.

System Architecture



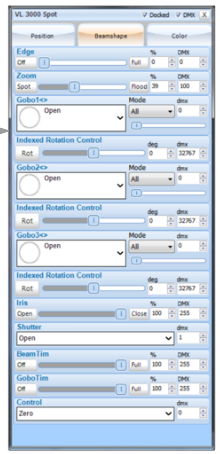
Module 6.2

System Configuration

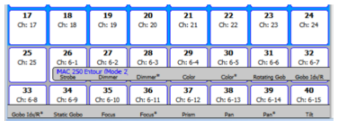


Types of equipment

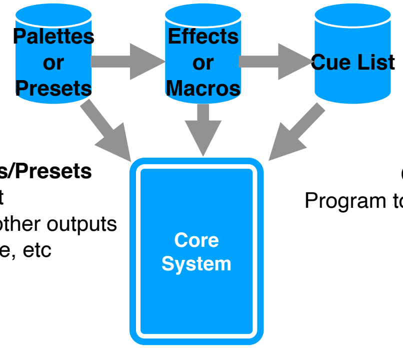
Relay - 1 Slot, 2 DMX values, e.g.<128 Off; >128 On
Lamp - 1 Slot, 255 DMX values; 0 = Off 255 = Full On etc
RGB Lamp - 3 Slots, 255 DMX values; Red, Green, Blue
etc



Inventory/Patch
What type of equipment?
What DMX start address?
Where is it located?



Programming

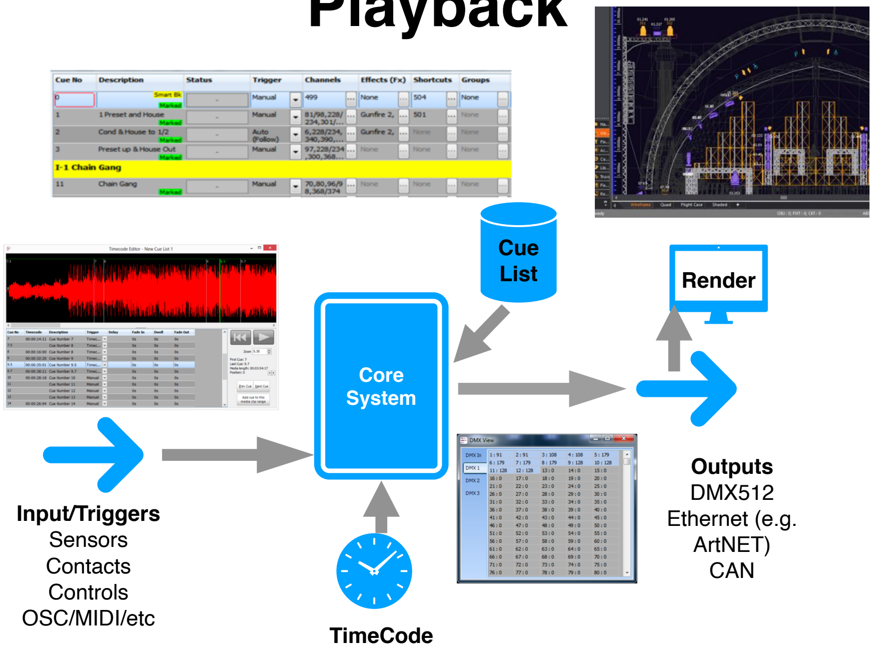


Shortcuts/Palettes/Presets
Presets for equipment
- e.g. certain lights, other outputs
- colour, location, size, etc

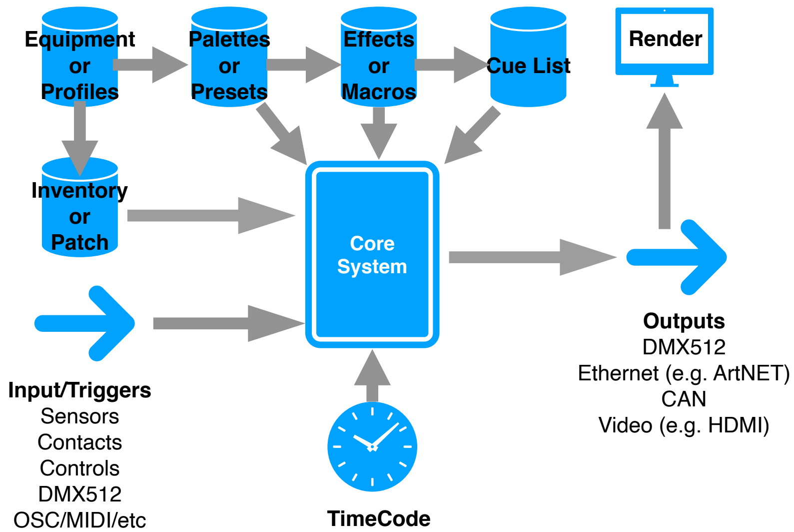
Cue List
Program to respond to inputs

Macros
- Sequences of presets
- e.g. to form a chase effect
- e.g. to track movement
- e.g. to automate changes in location/settings etc
- best thought of as a function/procedure in programming

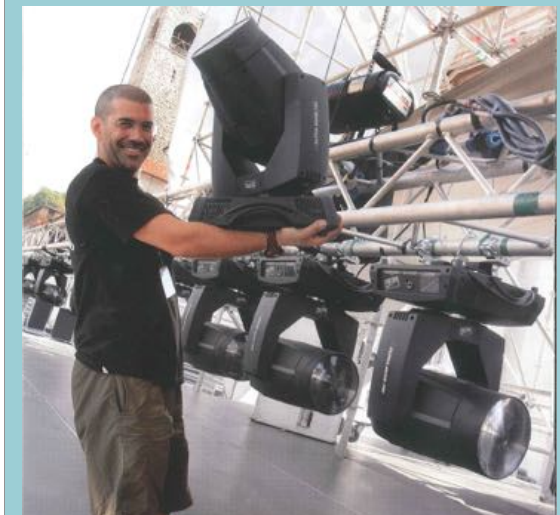
Playback



System Architecture

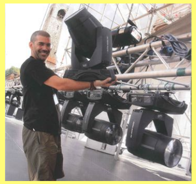


MULTI-SLOT FIXTURES



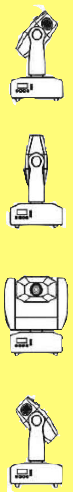
Module 6.3

MULTI-SLOT CONTROL



- Many receivers need more than one slot of control data
- Receiver needs to ensure the set of slots is consistent (use a flag to indicate if data is ready)

PAN/TILT SLOT VALUES



Device is tilted using stepper motors (or servos)

A “simple” device uses a 8-bits for each value:

Maximum 255 (i.e. 1.4 degrees /step for 360 degrees)

Often use a 16 bit value (a pair of two *consecutive* slots)

Maximum 64 k value (i.e. 0.005 degree /step)

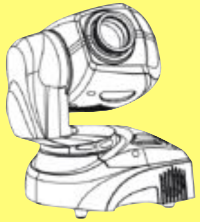
A	Pan

1.4 degrees /step

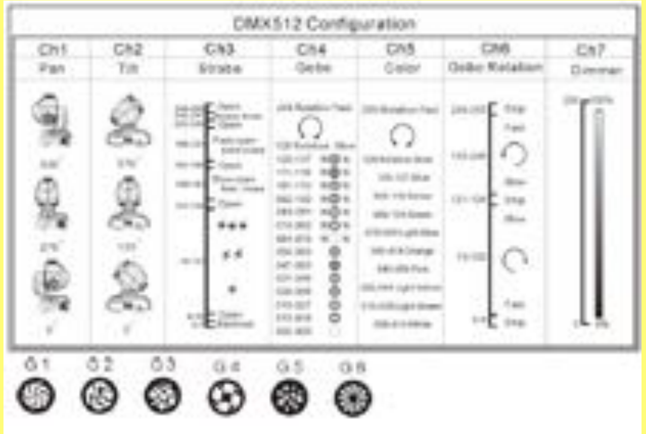
A	Pan coarse
A+1	Pan fine

0.005 degrees /step

GENESIS MS20 LED



Block of 8 DMX Channels
8 Colours LED PWM, Moving Head
Geared stepper motors

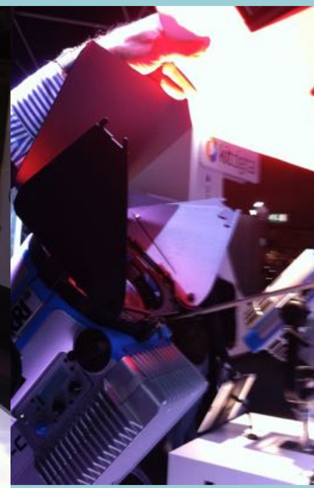
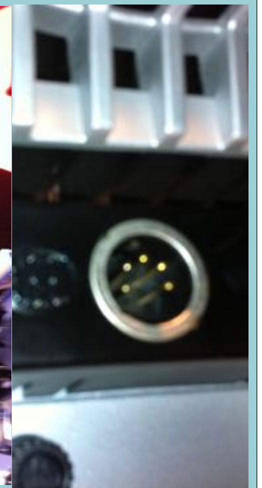


ROBOT

Industrial robot available for the film industry

Developed from a car assembly line robot

LEDS AND PWM

LED DRIVERS

LEDs are non-linear: Power supply circuits for LEDs need to avoid thermal runaway - when LED junction heats, the LED junction resistance decreases - as they heat they draw more power!

Simplest LED circuit uses a series ballast resistor
(significant for high power LEDs)
e.g. Vf =3.7V, I=300 mA

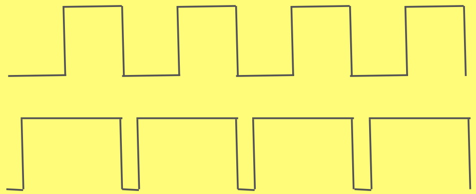
$$R_{ballast} = \frac{V_{in} - V_{forward}}{I_{LED}}$$

Care is therefore needed to limit current for high power LEDs

However, voltage drop across the ballast resistor wastes power!

A constant current source is a better solution for high power

DIMMING LEDS: PWM



128 = 50%

230 = 90%

Uses a MOSFET in series with the LED string

Pulse Width Modulation used to control ***power*** of LED Lamp

Receivers interpret DMX slot value as ***Pulse Width Ratio***

Pulses typically repeat at **kHz** rates for LEDs (re.g. 4kHz)

START CODES

Module 6.5

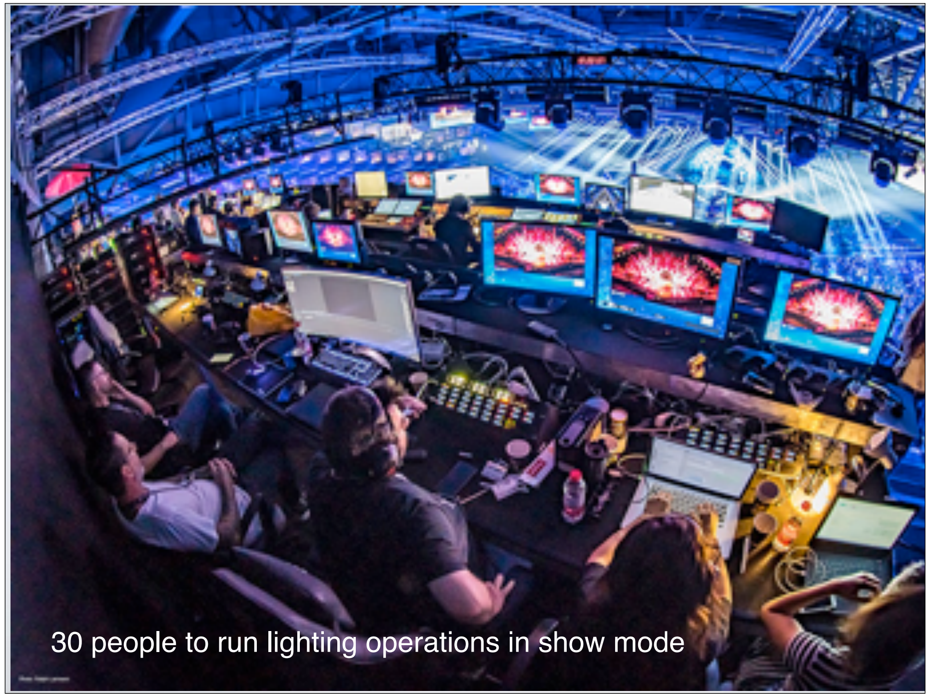
NETWORK TEST PACKET

Start Code = 0x55
All 512 data slots also carry value 0x55

A test frame be sent at any time.
It travels to all parts of the "universe".
It can be received by any DMX tester.
This can be used to discover any cable/repeater faults.
The start code 0x55 cause all ***normal*** receivers to ignore the frame

EUROVISON 2019

120,000 control parameters
250 active DMX universes.
80,024 meters of cable for lighting alone.



30 people to run lighting operations in show mode

IDENTIFICATION OF UNIVERSE

If there is only one controller, it's easy to plug into the correct cable bus.

As systems became more complex, people needed multiple buses. How do you know which receiver plugs into which cable?

We call each set of cables and equipment a "UNIVERSE". Universes can be numbered.

MFID PACKET & SI PACKET

0x91 (145)   MFID packet

first two slots contain a 16-bit Manufacturer-ID, remaining slots with proprietary data

0xCF (207)   System Information packet (SIP)

- normally 24 slots containing various data in pre-defined fields
  - slot 1: Slot count (a.k.a. SIP Checksum Pointer) [default is 24]
  - slot 2: Control Bit Field
  - slot 3 & 4 Checksum of Previous Packet
  - slot 5: Sequence Number
  - **slot 6: DMX Universe Number**
  - slot 7: DMX Processing Level
  - slot 8: Software Version
  - slot 9 & 10: Standard Packet Length (a.k.a. Universe Size)
  - slot 11 & 12: Number of Packets sent since previous SIP
  - **slot 13 & 14: Originating Device's MFID**
  - slot 15 & 16: 2nd Device's MFID ...
  - slot 21 & 22: 5th Device's MFID

Universe identifies the bus

ID identifies

CHECKING RECEIVE DATA

Send data frame(s) (SC 00) followed by SI Packet (SC 207)

SI Packet contains data about the UNIVERSE

SIP identifies the Universe number

Can identify which equipment ***sent frame***

Can verify ***no SI Packets were lost*** (sequence number)

Count of how many frames since last SI Packet

Can verify ***no Data Packets were lost***

Count of how many frames since last SI Packet

Count of how many bytes per data frame (standard length)

MFID PACKET & SI PACKET

Also contain integrity check….

0xCF (207)   System Information packet (SIP)

- normally 24 slots containing various data in pre-defined fields
  - slot 1: Slot count (a.k.a. SIP Checksum Pointer) [default is 24]
  - slot 2: Control Bit Field
  - slot 3 & 4 **Checksum of Previous Packet**
  - slot 5: Sequence Number
  - slot 6: DMX Universe Number
  - slot 7: DMX Processing Level
  - slot 8: Software Version
  - slot 9 & 10: Standard Packet Length (a.k.a. Universe Size)
  - slot 11 & 12: Number of Packets sent since previous SIP
  - slot 13 & 14: Originating Device's MFID
  - slot 15 & 16: 2nd Device's MFID ...
  - slot 21 & 22: 5th Device's MFID

Integrity check for previous frame

Cirque du Soleil

SUMMARY

Code	Meaning	Notes
0000 0000	Lighting Control Data	Default format
0101 0101	Network Test	All slots carry the same value
0001 0111	Text Packet	Simple text message
1100 1100	Remote Device Management	RDM Control/Response message
1100 1111	System Information Packet	Identifies a DMX Universe
1111 1111	Dimmer Curve Select	

https://tsp.esta.org/tsp/working_groups/CP/DMXAlternateCodes.php

Control for Electromagnets

Control for Electromagnets

PROTECTING DATA

Send data frame (SC 00) followed by SI Packet (SC 207)

SI packet contains a CRC to detect errors ***within the SI Packet***

Can verify which equipment ***sent frame***

Can verify ***no SI Packets were lost*** (sequence number)

An SI packet also carries a CRC that covers the ***last data frame***

Only frames protected by a SI Packet are accepted by a receiver for a critical control application

HIGHER ASSURANCE (1)

How can we use what we know to make a safe design?

Receiver needs to be designed to have a very low chance of accepting a corrupted frame.

Here is one way:

Normally the receiver is disabled

The first step explicitly activates the receiver for a short period of time (called “arming”)

The second step sends a command to the armed receiver

All frames are protected by CRCs.

HIGHER ASSURANCE (2)

Send a sequence of 4 frames:

Frame (SC 00) to “ARM” receiver 4.5-5 seconds before use
Followed by SI Packet (SC 207), protecting the “ARM”
Frame (SC 00) with slots to “FIRE” an “ARM”ed receiver
Followed by SI Packet (SC 207), protecting the “FIRE”

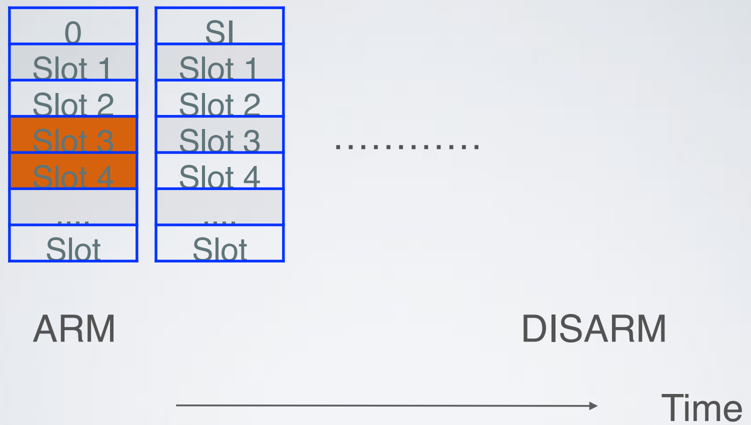
Receiver:

Only accepts frames followed by a valid SI Packet.

Only accepts a “FIRE” when “ARM” previously received *within* 4.5-5 seconds, otherwise it disarms itself.

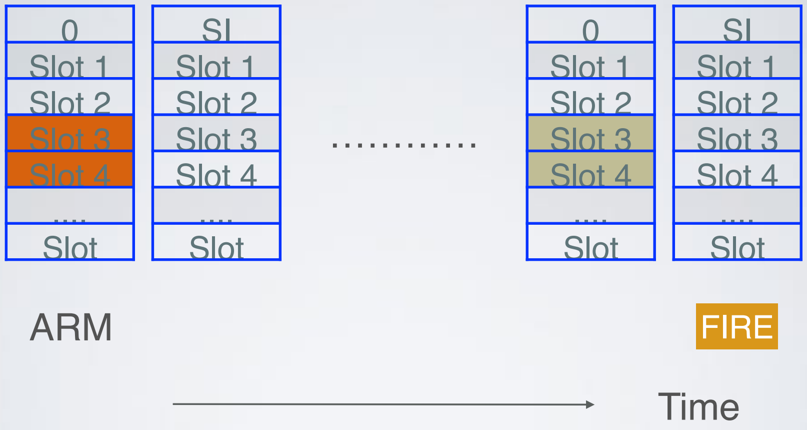
Some “visible” indicator could show the “armed” units, allowing an operator to cancel the “fire” command if not appropriate.

HIGHER ASSURANCE (3)

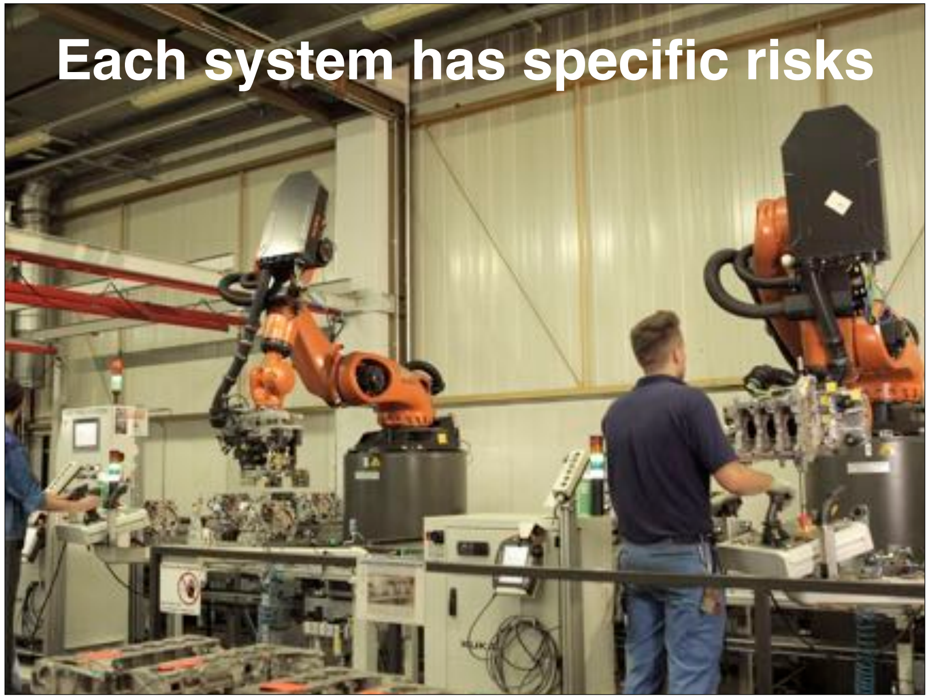


Receiver armed ***only when next SIP says it is valid***
Fails safe if no command received

HIGHER ASSURANCE (4)



Receiver armed then receives the Fire Command
If both verified correctly the action is taken!



REPEATERS & REGENERATION

Module 7.1

LARGER APPLICATIONS

Digitally ***regenerates*** the signal

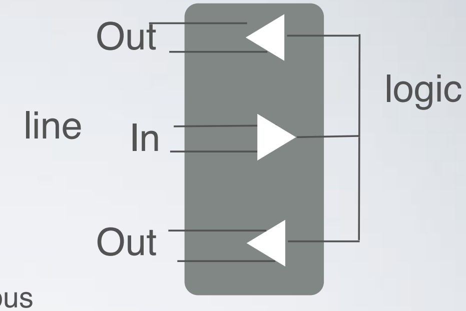
- All parts of the "Universe" see the ***same 512 DMX Slots***

Enables:

- Run cables > 300m

- Connect more than 32 devices within a single "DMX Universe"

SPLITTERS/REPEATERS



Splitters regenerate signal

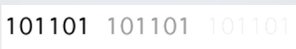
Simple for a unidirectional bus

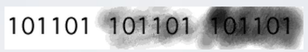
Ground isolation (transmitter grounded)

Many splitters usually allow user to isolate specific output

More complicated for bi-directional links (e.g. RDM)

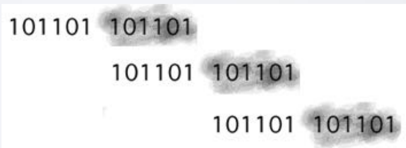
REGENERATION - NOT JUST FOR DR WHO

- Attenuation: 

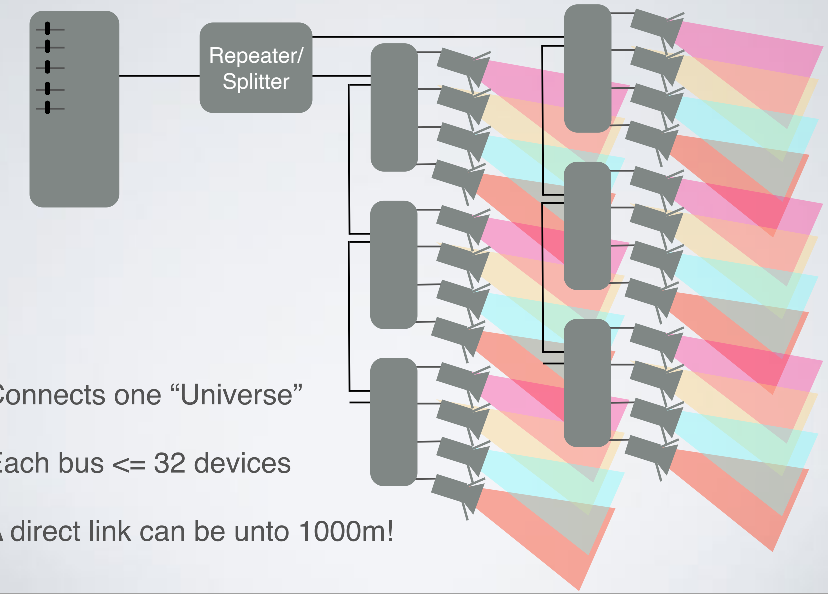
- Noise: 

- Distortion: 

- Result is signal degrades with ***distance***
- Regenerative repeaters enable operation at a ***distance***



SPLITTERS/REPEATERS



Connects one "Universe"

Each bus <= 32 devices

A direct link can be unto 1000m!

SENDING OVER ETHERNET TRANMISSION

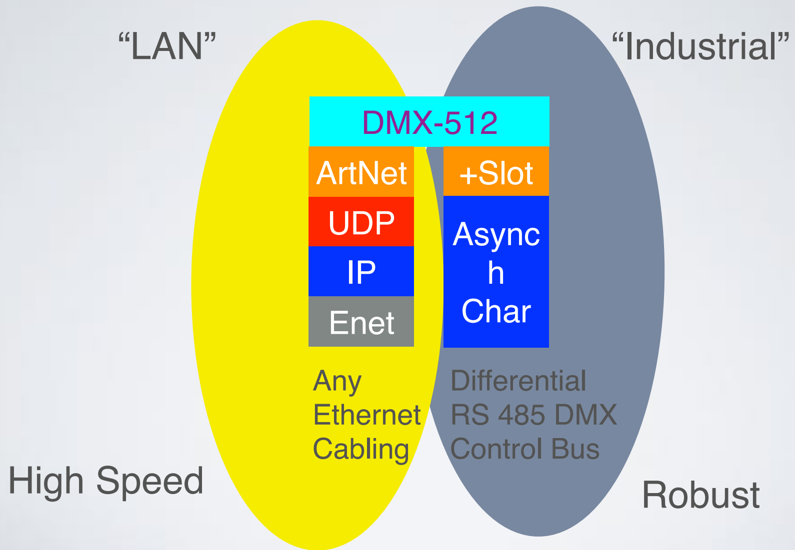
Module 7.2

DMX OVER ETHERNET

DMX frames can be transported over Ethernet (e.g. ArtNet)

- PCs and Phones can run programs to read/write DMX
- DMX data is sent as UDP datagrams using IP
- Can be accessed anywhere in the world over the Internet
- Works over standard CAT5/CAT6 copper cable and fibre

PROTOCOL STACK



ETHERNET CONTROL

EIA-485 is suited to industrial control

Higher Noise Immunity

Longer Cables

Robust connection and fail-safe communications

DMX/Ethernet also has advantages

Office/Computer equipment often has Ethernet Interfaces

Most buildings are already cabled for Ethernet

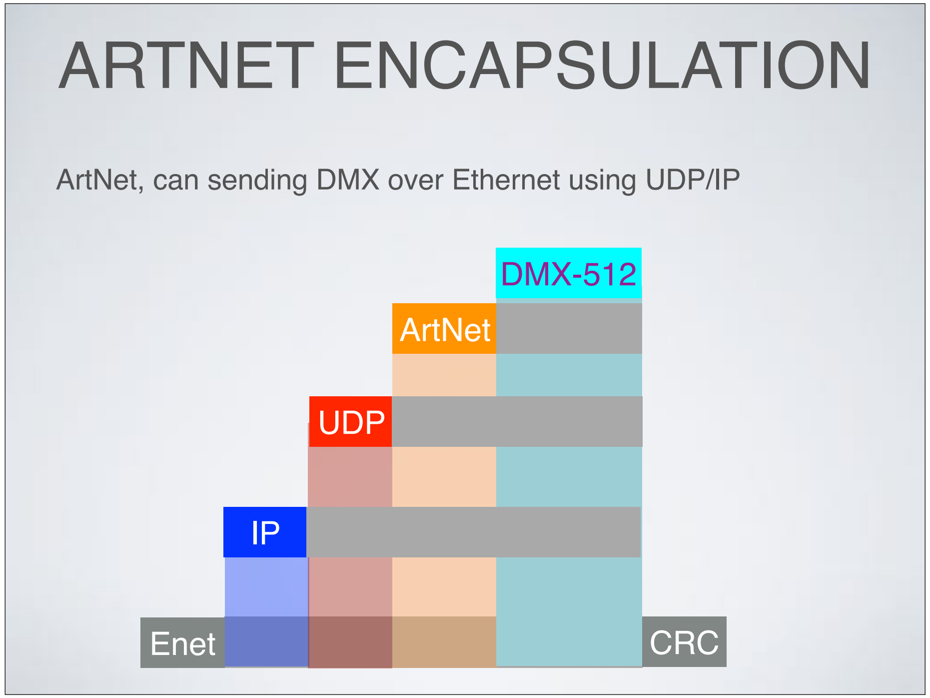
A single Ethernet cable can carry many DMX Universes

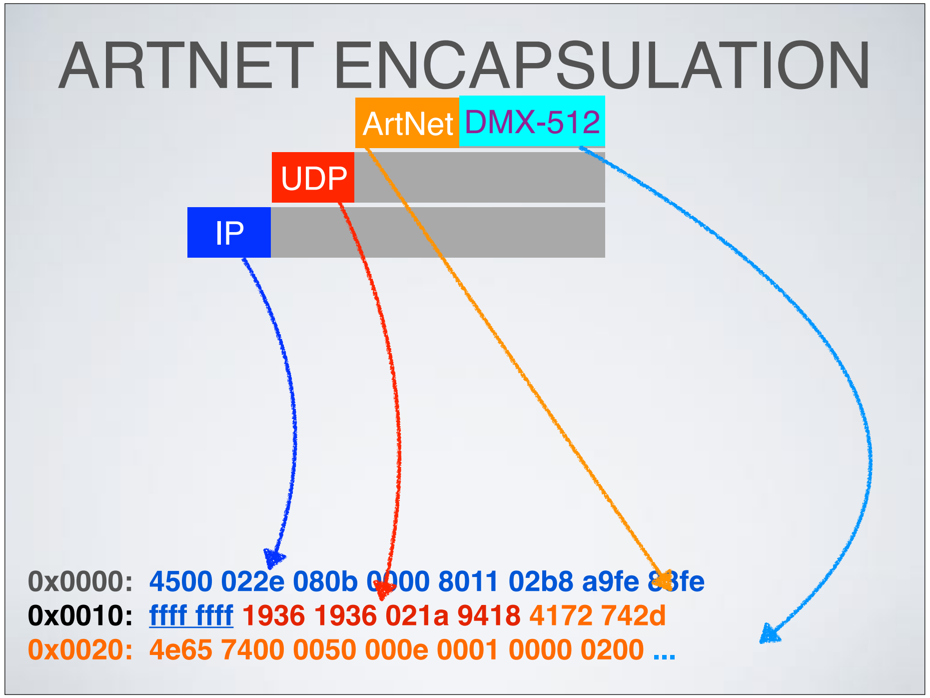
DMX/Ethernet has disadvantages

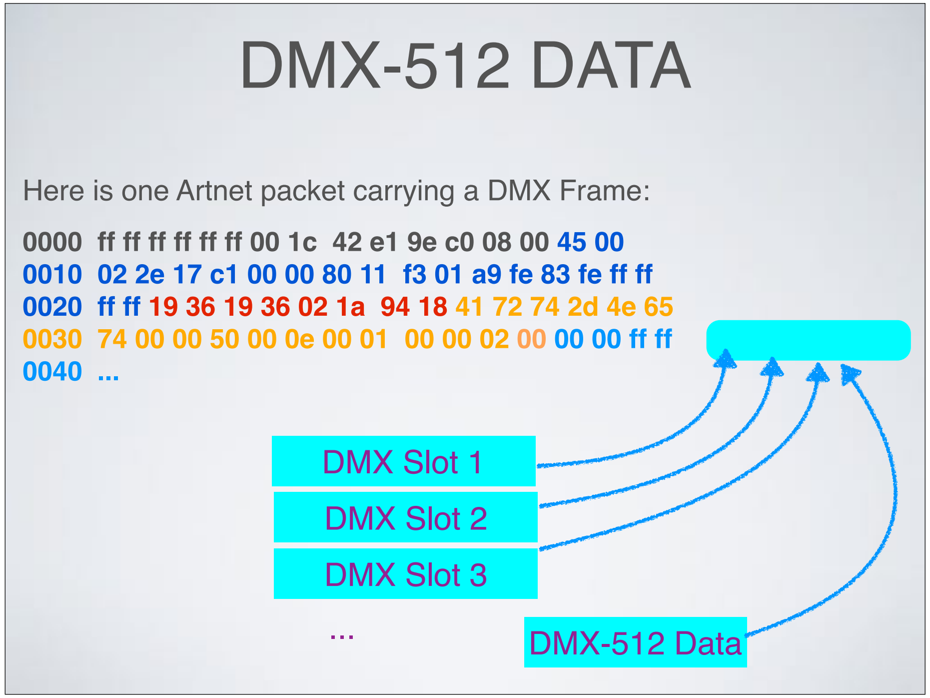
Not designed for industrial use (unshielded)

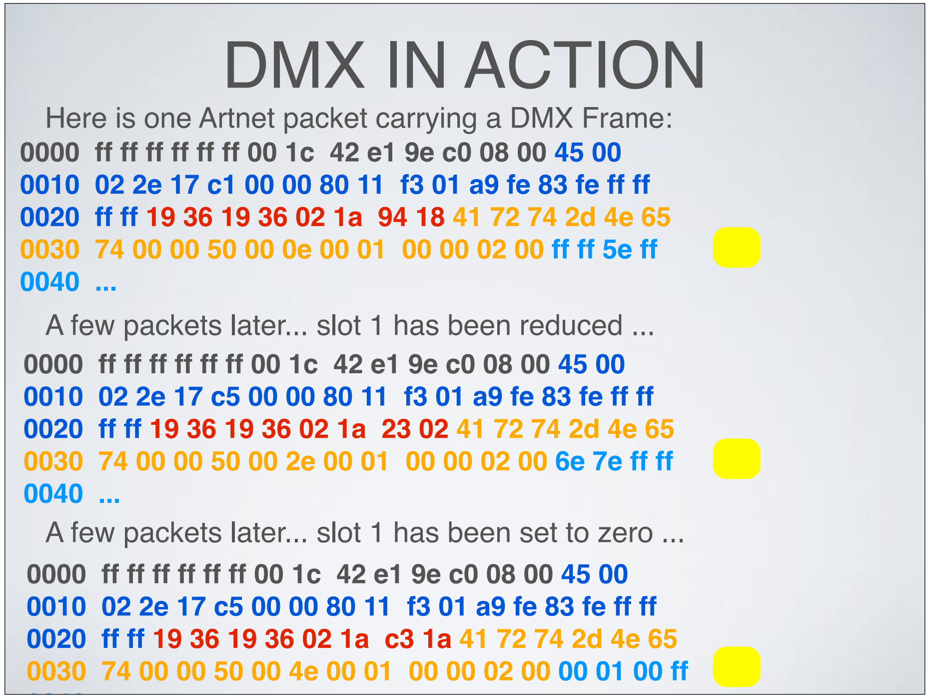
Twisted pair cables restricted to <100m in length

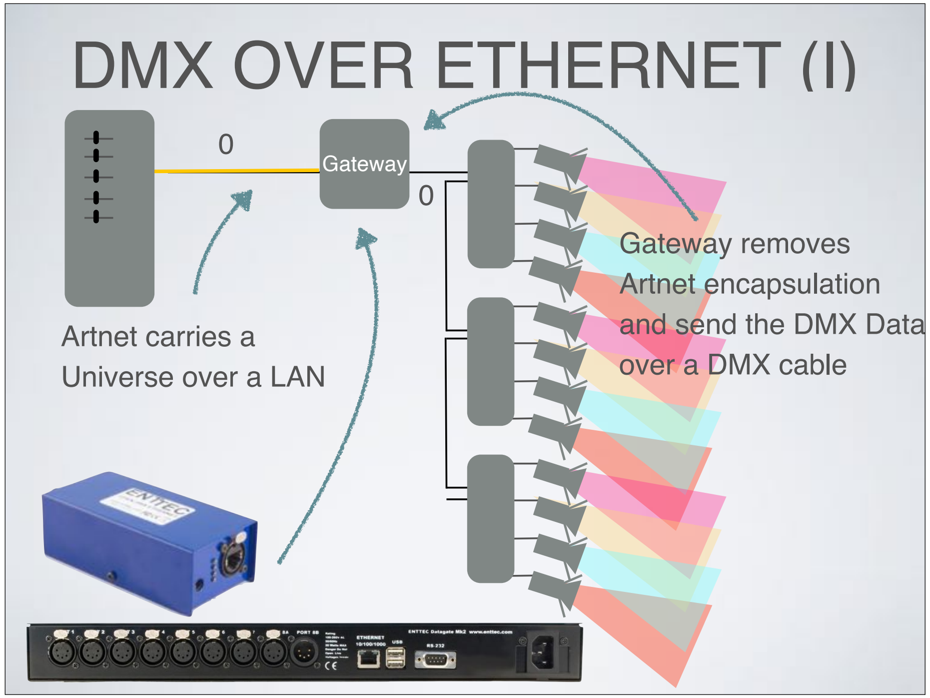
Less robust RJ-45 connectors, easily broken

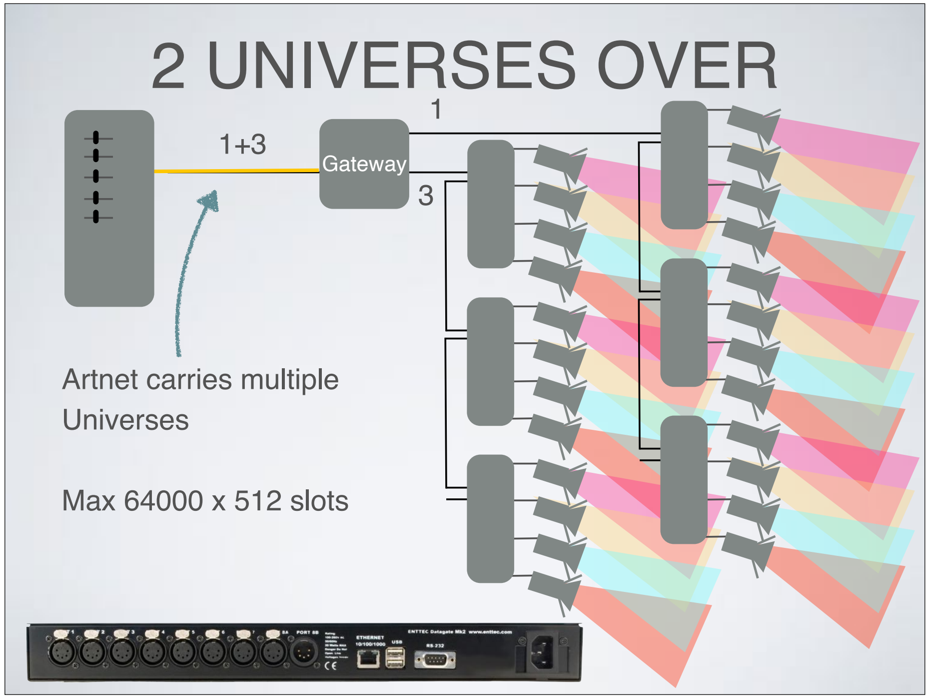

ARTNET ENCAPSULATION
ArtNet, can sending DMX over Ethernet using UDP/IP
DMX-512
ArtNet
UDP
IP
Enet
CRC

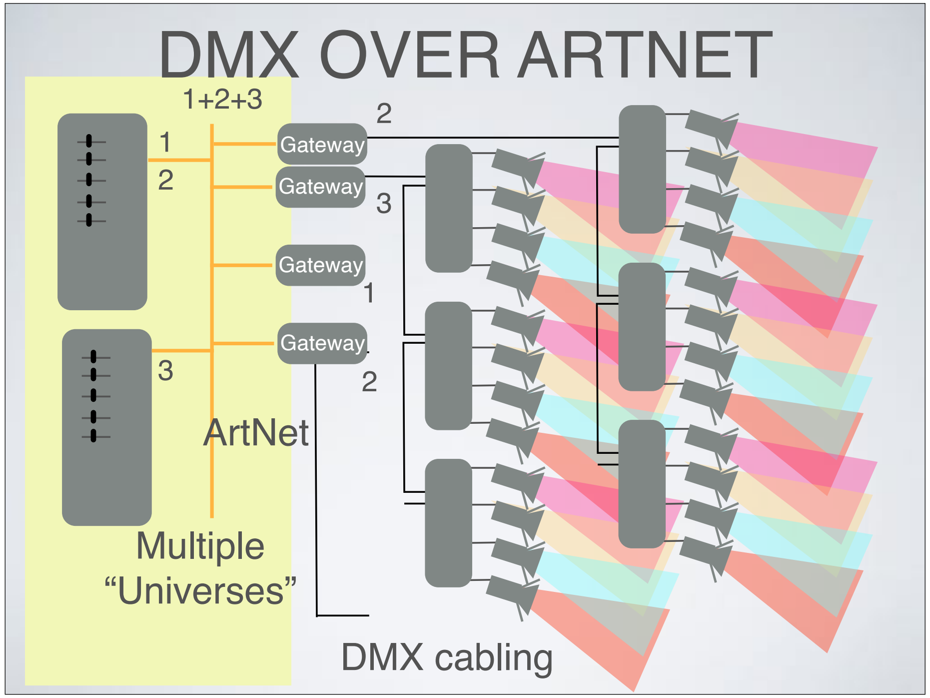

ARTNET ENCAPSULATION
ArtNet
DMX-512
UDP
IP
0x0000: 4500 022e 080b 0000 8011 02b8 a9fe 83fe
0x0010: ffff ffff 1936 1936 021a 9418 4172 742d
0x0020: 4e65 7400 0050 000e 0001 0000 0200 ...

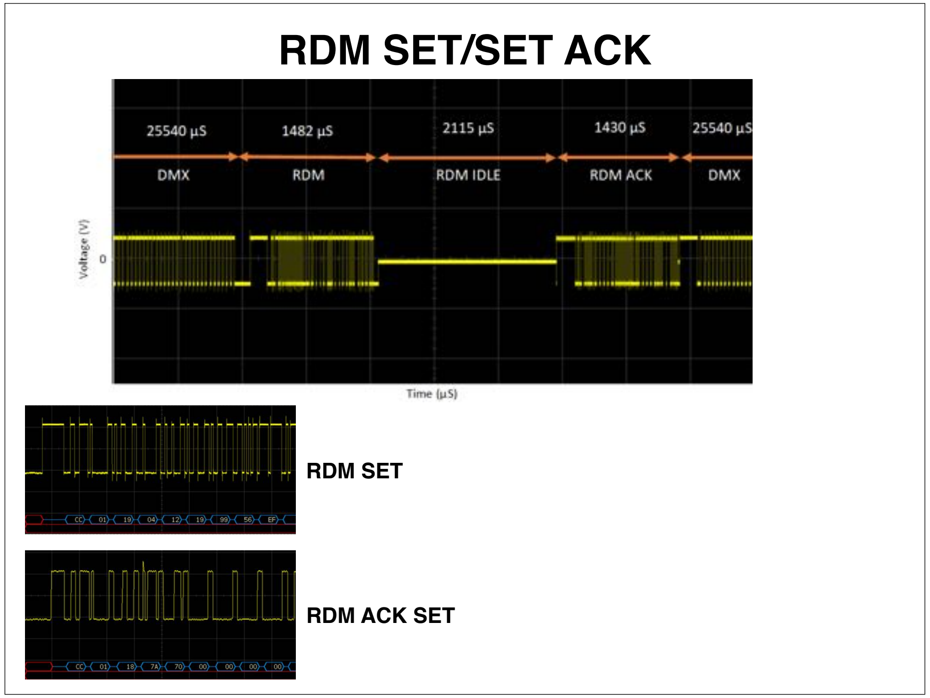

DMX-512 DATA
Here is one Artnet packet carrying a DMX Frame:
0000 ff ff ff ff ff ff 00 1c 42 e1 9e c0 08 00 45 00
0010 02 2e 17 c1 00 00 80 11 f3 01 a9 fe 83 fe ff ff
0020 ff ff 19 36 19 36 02 1a 94 18 41 72 74 2d 4e 65
0030 74 00 00 50 00 0e 00 01 00 00 02 00 00 00 ff ff
0040 ...
DMX Slot 1
DMX Slot 2
DMX Slot 3
...
DMX-512 Data


DMX IN ACTION
Here is one Artnet packet carrying a DMX Frame:
0000 ff ff ff ff ff ff 00 1c 42 e1 9e c0 08 00 45 00
0010 02 2e 17 c1 00 00 80 11 f3 01 a9 fe 83 fe ff ff
0020 ff ff 19 36 19 36 02 1a 94 18 41 72 74 2d 4e 65
0030 74 00 00 50 00 0e 00 01 00 00 02 00 ff ff 5e ff
0040 ...
A few packets later... slot 1 has been reduced ...
0000 ff ff ff ff ff ff 00 1c 42 e1 9e c0 08 00 45 00
0010 02 2e 17 c5 00 00 80 11 f3 01 a9 fe 83 fe ff ff
0020 ff ff 19 36 19 36 02 1a 23 02 41 72 74 2d 4e 65
0030 74 00 00 50 00 2e 00 01 00 00 02 00 6e 7e ff ff
0040 ...
A few packets later... slot 1 has been set to zero ...
0000 ff ff ff ff ff ff 00 1c 42 e1 9e c0 08 00 45 00
0010 02 2e 17 c5 00 00 80 11 f3 01 a9 fe 83 fe ff ff
0020 ff ff 19 36 19 36 02 1a c3 1a 41 72 74 2d 4e 65
0030 74 00 00 50 00 4e 00 01 00 00 02 00 00 01 00 ff


DMX OVER ETHERNET (I)
0
Gateway
0
Gateway removes
Artnet encapsulation
and send the DMX Data
over a DMX cable
Artnet carries a
Universe over a LAN


2 UNIVERSES OVER
1+3
Gateway
1
3
Artnet carries multiple
Universes
Max 64000 x 512 slots


DMX OVER ARTNET
1+2+3
1
2
3
Gateway
Gateway
Gateway
Gateway
2
3
1
2
ArtNet
Multiple
“Universes”
DMX cabling


RDM SET/SET ACK
25540 µS
1482 µS
2115 µS
1430 µS
25540 µS
DMX
RDM
RDM IDLE
RDM ACK
DMX
Voltage (V)
0
Time (µS)
RDM SET
RDM ACK SET