Imperial College London

Topic 6

Electrical Signals

Peter Cheung
Dyson School of Design Engineering

URL: www.ee.ic.ac.uk/pcheung/teaching/DE1_EE/
E-mail: p.cheung@imperial.ac.uk

This lecture is about the different types of electrical signals. The content of this lecture is important for you to understand Lab 1.

Remember, lab works often involve making mistakes and finding them. To minimize these mistakes and the resulting frustrations, you must READ THE INSTRUCTION carefully, and check that what you build on the breadboard is what is written down (i.e. no connection errors, particularly those related to power and ground wires).

Electronic Logbook

- You must keep a **logbook** because:
  - It is a professional practice.
  - It helps you to learn; forces you to think and to reflect.
  - It is useful for your future reference.
- Here are some guidelines on using the logbook:
  - Logbook should include sketches, your thoughts etc., and does not need to be tidy. It definitely should not be "beautiful".
  - Never copy someone else's logbook – it is both cheating and useless to you.
  - Attach photo's, plots, calculations, codes or anything else interesting in your logbook.
  - At the Oral Assessment in the last week of term, you will be asked to show the examiner your logbook and asked questions where answers are only be found in it.
- Electronic Logbook
  - MS Word or similar – simple, no learning required, sequential only
  - MS OneNote – Easy to learn, has hierarchy: book, section, pages
  - Github – Combines logbook with version control; probably more suitable for software lab than hardware lab.

PYKC 12 May 2020 DE1.3 - Electronics Topic 6 Slide 2

As a "bonus" teaching of my module, you will learn how to keep a good electronic logbook.

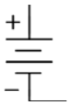
The purpose of keeping a logbook is much more than having a record (or a log) of what you have done so that you can refer to it in a later date. When you conduct experiments or a project, or if we write some computer program or build some electronic hardware, there is a strong tendency for you to "hack" or try to achieve results through trial-and-error. If you develop a habit of keeping a logbook, you will find that this process forces you to slow down and THINK more, reflect on what you have done and more likely to make plan on what to do next. The over result is that you will learn more and better.

While in the past, we encourage our students to keep a paper logbook, I am encouraging you to use an electronic logbook. This means that you would need to have a laptop with you. There are various application programmes that may help you. For this module, I encourage you to try OneNote by Microsoft. This is free for Imperial College and is very easy to earn.

Please read my one-page guideline on how to keep a good electronic logbook, which you can find on the course webpage. In it I provide a link to an excellent video tutorial to OneNote.

Electrical signals – DC vs AC

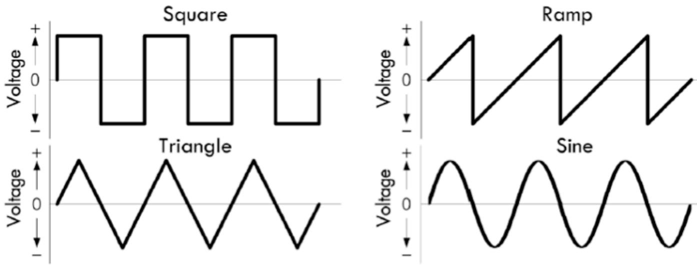
- Electrical voltage or current can be constant over time (at least over a reasonably long period of time) or vary over time.
- There are various types of electrical signals (voltage or current):



1. **Constant or dc/DC** - For example, the output voltage of a fully charged battery is substantially constant. Connecting a resistor across the battery will also result in a constant current flowing from the battery to the resistor. We call this dc voltage or current (i.e. **direct current**).

   $V = k$, where k is a constant.

2. **Alternating Current or ac/AC** - Voltage from the mains supply is changing over time, and it alternates between positive and negative value following a sinusoidal function. We call this ac voltage (i.e. **alternating current**).



P84

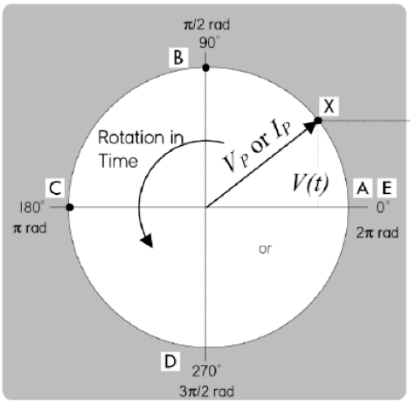
Let us first consider a direct current or **DC signal**. This is hardly a signal because it is a constant voltage or current. It is in fact a **source**. All real sources have some **source resistance**. For a realistic voltage source such as a battery, the resistance is in **SERIES** with the source. For a realistic current source (which we don't encounter often in day-to-day life), the resistance is in **PARALLEL** with the current source.

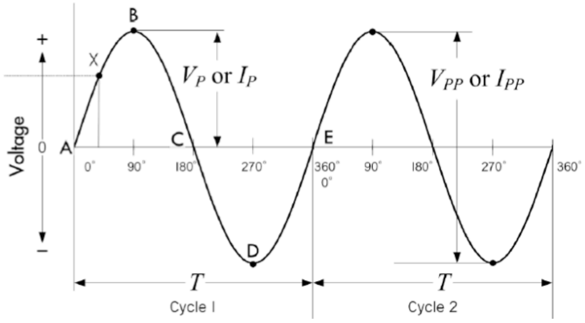
The second important signal is the alternating or **AC signal**. This is probably the most important signal in electronics, communications, signal processing and control. **Sine and cosine signals** are special cases of AC signals. All it means is that the AC signal changes in time.

The **mains** supply to which we plug in our equipment is a power source with a sinusoidal signal at 50Hz (in the UK) and approximately 230V. We will consider what the value 230V means exactly later.

A perfect sinewave is completely characterized by only three parameters: its amplitude, its frequency and its phase. Again we will consider sinewaves in more details later.

AC signal – mathematical representation





$$V(t) = V_P \sin\ (2\pi f t + 0^o) \qquad f = 1/T$$

- A sinewave is characterized by its:
  1. Amplitude Vp (or Ip for current)
  2. Frequency f in Hz (or ω in rad/sec) – remember: $\omega = 2\pi f$
  3. Phase Φ

P85-86

All AC signals are characterized by three parameters. The example shown here is for a sine wave signal. This also applies to a square wave or a triangular wave signal.

The size of an AC signal is determined by its amplitude Vp, which is the voltage magnitude from the mean value (in this case, the mean value is 0V) to the maximum or minimum. (Magnitude means the absolute value.) Sometimes, we measure the amplitude between it maximum and minimum values. We called this amplitude peak-to-peak voltage Vpk-pk.

An AC signal is ALWAYS periodic. The repetitive interval is the PERIOD T and the frequency in Hz is 1/T. However, we often define frequency as angular frequency ω in radians per second.

Finally, the phase angle Φ specify the starting angle (or phase) of the sine wave at time = 0.

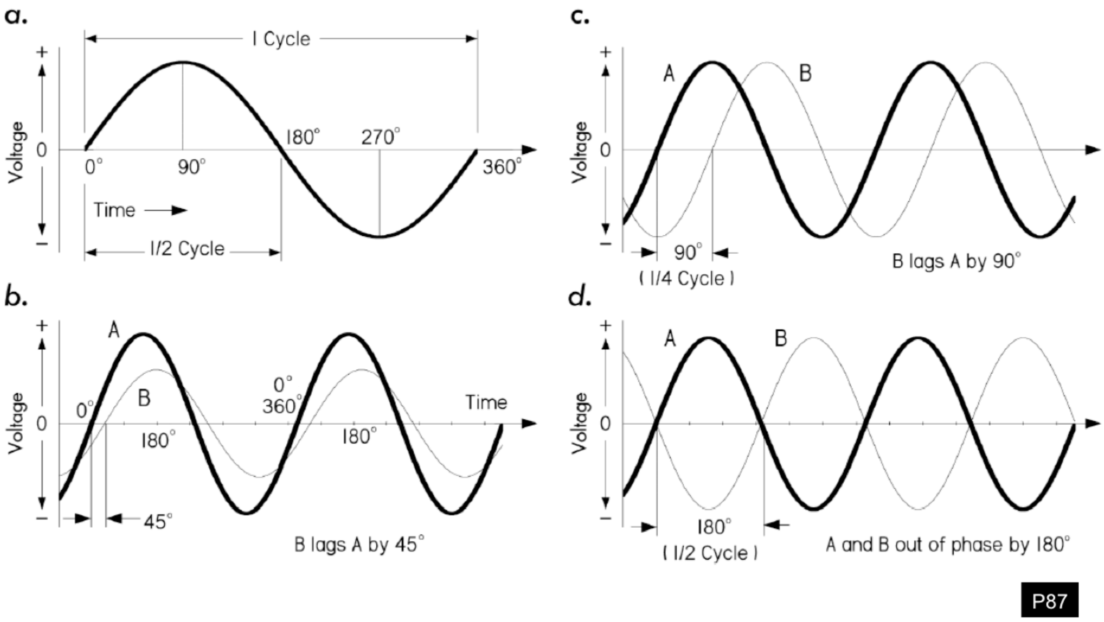
A sine wave is often represented by a vector of length Vp rotating around the origin at a rate of f rotations per second as shown in the slide. One completion rotation is a cycle and it goes through 360° or 2π rad. Hence $\omega = 2\pi f$.

The sine wave signal at any given time V(t) is the projection of this vector on the x-axis.

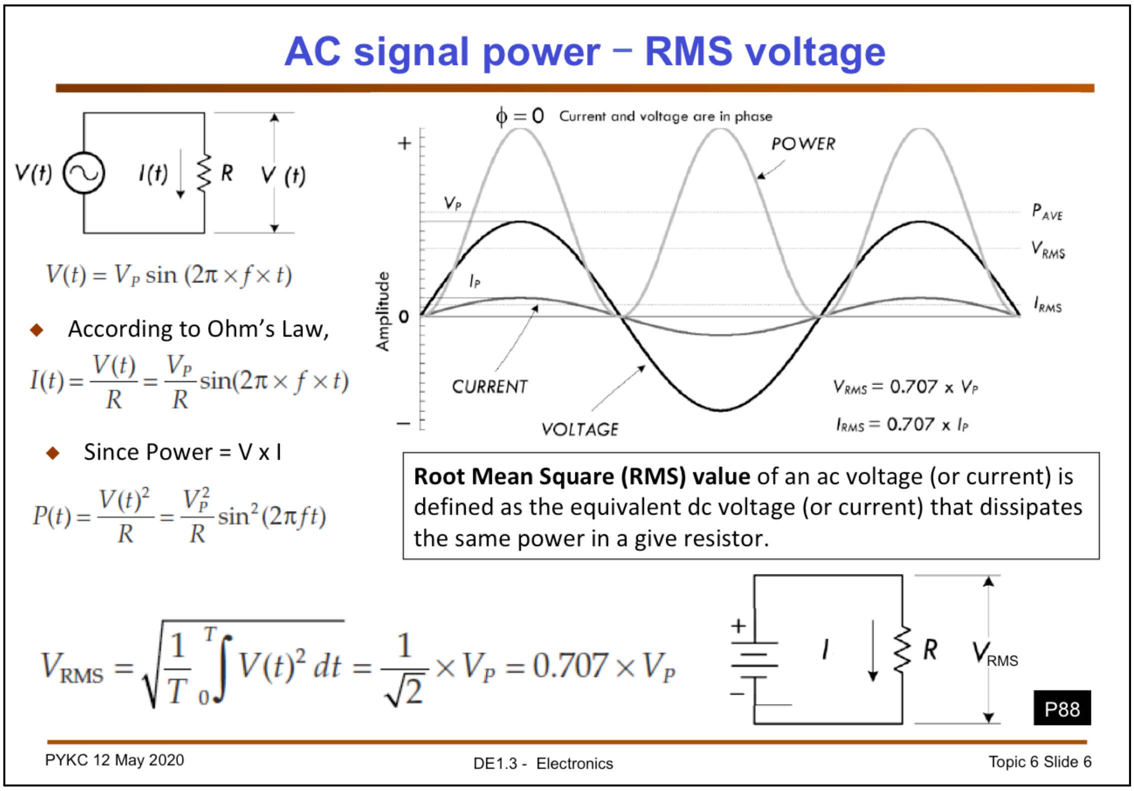
Finally the phase is the starting angle of the vector at t = 0. Hence the equation of a sine wave is:

$$V(t) = V_P \sin\ (2\pi f t + 0^o) \qquad f = 1/T$$

AC signal – relative phases in sine waves



Here we show two sine wave signals at the same frequency but with different phases relatively to each other.



An ac signal is time varying. How can we define a single quantity that measures the SIZE of an ac voltage V(t)?

We cannot use the peak value alone, because the peak value does not take into account the shape of the waveform. We cannot use its average voltage or current, because an average of an ac signal is ..... ZERO - the positive and negative of this symmetrical signal cancel each other out!

So we define a new quantity which encapsulate the "size" based on the power dissipation of such a signal in a resistor. The Root Mean Square voltage of an ac signal is the equivalent dc voltage V_{RMS} that dissipate exactly the same power as the ac signal. In other words, both signals when applied to a given resistor will raise it to the same temperature.

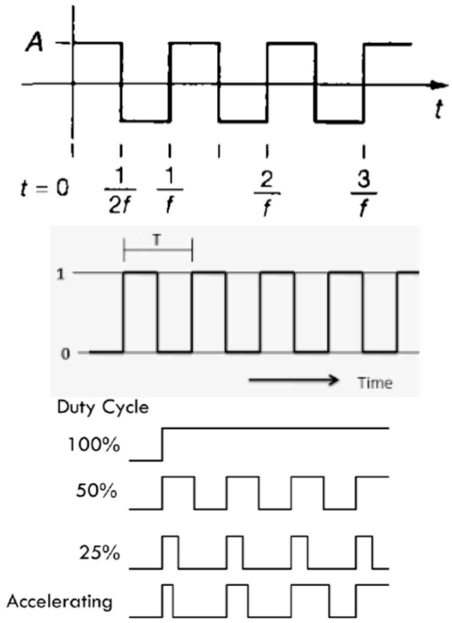
The power dissipated by a dc voltage V_{RMS} when applied to a 1Ω resistor is V_{RMS}^2. The dissipation by the ac voltage is found by computing the average value of $V(t)^2$.

Hence:

$$V_{\mathrm{RMS}} = \sqrt{\frac{1}{T}\int_0^T V(t)^2\,dt} = \frac{1}{\sqrt{2}} \times V_P = 0.707 \times V_P$$

Square wave, digital and PWM signals

- A square wave signal is symmetrical around 0 and have equal period at +A and –A.
- A digital signal is similar to square wave, but it only goes between 0 and +A. For example, for the ESP32 microcontroller, +A = 3.3V.
- A special type of digital pulse signal is known as a symmetric **clock signal** where the time spent in +A and in 0 are the same – e.g. the clock signal used in most microprocessors.
- In general, the duration spent on +A (mark period) and 0 (space period) may be different. This ratio mark:space (mark/space ratio) may also be expresses as **duty cycle** in percentage.
- Duty cycle = (mark period/total period) x 100%.
- A digital signal with varying duty cycle is also known as **Pulse Width Modulation** (PWM) signal.



PYKC 12 May 2020 DE1.3 - Electronics Topic 6 Slide 7

A **square wave** is obvious. Normally a square wave is assumed to be symmetrical in amplitude (going between ±A) and spend equal time in +A and –A.

More common is the pulse signal or repetitive pulse signal. A pulse signal or sequence of pulses are used in many situation – such as an ultrasound distance sensor would use a burst of pulses to measure distance through echo location.

Even more common is the use of repetitive pulse signal as a **clock signal** in digital circuits. Digital signals go between 0 and A. A symmetrical clock signal spend equal amount of time in A (or high level) and in 0 (or low level).

In general a repetitive digital signal may not has the same duration on 'high' and 'low'. The ratio of time on high (mark level)/ time on low (space level) is known as **mark-space ratio**. Therefore a symmetrical clock signal has a mark-space ration of 1:1.

Another common way of describing a repetitive digital signal is its duty cycle. This is defined by the percentage of time spent on the high level in a period. A signal that has its duty cycle varying over time is called a Pulse Width Modulation signal. PWM signals are generally what we use to control the brightness in an LED light or the speed of a motor. We will be considering how we use PWM to control motors and LEDs later in Lab 4.

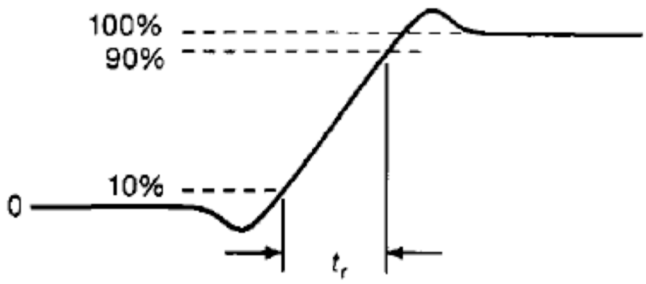
Noise signal

- Noise signal – general this is the signal voltage or current that you DON'T want, but it is there anyway. The noise signal can be random (such as noise signal from space), or predictable (e.g. noise caused by picking up the electrical mains at 50Hz).

Here is another type of signal known as a noise signal. Noise is a general term, referring to the component of a signal that you do not want, but it is there! If you are measuring the mains supply voltage signal, you will see the 50 Hz sine wave. This signal is NOT noise – it is the one that you want to measure. However, if you are measuring your cardiac signal (ECG) you may find superimposed on your wanted signal a 50 Hz "hum" that you don't want. The 50 Hz component is now regarded as noise. In other words, what is noise and what is signal depends entirely on the context.

Step signal

- **Step signal** – this is a theoretical signal where a signal goes from P to Q instantaneously. The step can be positive (Q>P) or negative (P>Q). In practice, the signal will take finite time to transit. Rise (fall) time t_r is defined as the time when a step signal rises (falls) from 10% to 90% of the final level.



In real electronics, when a digital signal goes from 0 to 1, or zero volt to supply voltage (say 5V for TTL), it takes finite time for the signal to reach its final value. This is a **step signal** and is characterized by its **rise time**, which is defined by the time it takes to go from 10% to 90% for a rising edge (and 90% down to 10% on a falling edge for the **fall time**).

A step signal are often used and produced in the physical analogue world. For example, if you move a robot arm from A to B and monitor the signal from a sensor (transducer) that measures the arm position, the ideal case is see a perfect step signal with zero rise time. In reality, you are likely going to see a waveform similar to the one shown at top of this slide. There is not only a finite rise time, there could be an overshoot, or even oscillation that eventual dies down.

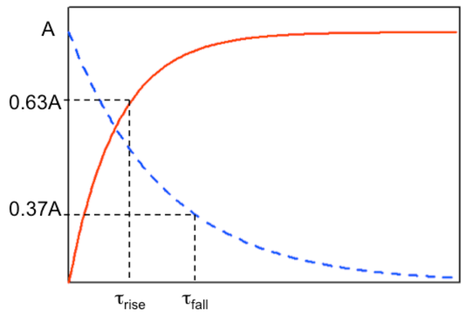
Exponential signals

- **Exponential signal** – This is often the output signal of a circuit driven by a step signal at the input. The mathematical representation of this signal is:

$V = A\,(1 - e^{-t/\tau})$ for rising exponential

$V = A\ e^{-t/\tau}$ for falling exponential

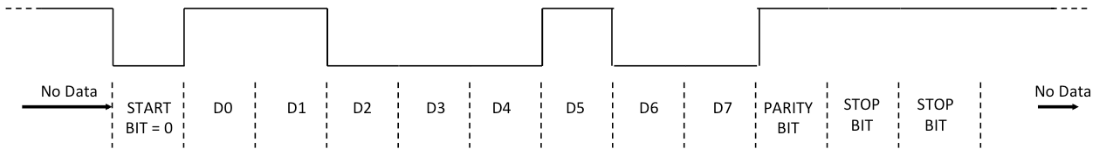
τ is often called time constant, and it is the time for the signal to reach 63% of the final value for a rising exponential and 37% for a falling exponential.



Next, we have the **exponential** rise and fall **signals**. This Is found everywhere, particularly in digital circuits. The rising exponential signal is characterized by its final voltage value A, and the time it takes to reach 63% of A. This time is known as the **TIME CONSTANT**.

Complex digital signal - UART

- Finally, in digital electronics, we often use digital signals to convey information by sending '1's and '0's at different times.
- One common example is a serial data standard known as Universal Asynchronous Receiver/ Transmitter (UART) signal.



- Normally, the signal is a logic '1' (or high voltage).
- It goes low for one clock period – this is known as the **START bit**.
- The signal then goes high or low depending on the value of **DATA bits**, least significant bit first. Therefore the 8-bit data represented by this signal is 00100011 (35 decimal or 23 hex).
- There follows another bit known as **PARITY bit**. This is option and may be absent (i.e. no parity). It may be ODD parity, which means that this bit is 1 or 0 such that the total number of '1's in the data stream + the parity bit is **ODD**. (Similarly for **EVEN** parity).
- Finally the data signal is high for either 1 or 2 bit periods – it is called the **STOP BIT**(s).
- 1/bit period (or bit frequency) is called **baudrate** (or number of bit periods per second).

The signal shown in this slide is called a "UART" signal (UART stands for **U**niversal **A**synchronous **R**eceiver **T**ransmitter). The data bit rate (also called baudrate) is typically 9600. Therefore every bit of the data lasts for 1/9600 = 104 μs.

UART has a specific data bit sequence. When there is no data, the logic level is '1'. When a character is sent, the first data bit period is 0. This is known as the START BIT (always '0'). It is then followed by 8 bits of data, least significant bit first. Then there could be an optional parity bit P. One could use even, odd or no parity bit. If no parity is used, the P bit is missed out. If you use odd parity, the number of ones including the data and parity bits is an odd number. For even parity, the total number of ones is an even number.

We are using EVEN parity in the waveform shown in the slide. Since the ASCII code of the letter '#' is hexadecimal 0x23, and there are three '1's in 0x23, therefore the parity bit is therefore '1' in order to make the total number of 1's an even number.

Finally it is guaranteed to have 1 or 2 bit period of high value. This is known as STOP BITS. We use 2 stop bits here.

ASCII codes

- Computers often communicate with the keyboard using serial data format such as the UART signal.
- The information carried by the signal in its data bit is coded using an old standard known as **A**merican **S**tandard **C**ode for **I**nformation **I**nterchange (**ASCII**).
- Therefore the example waveform shown in the previous slide is a UART signal that represent the character "#'.

Here is the ASCII table. You will find this useful as a reference. Even Matt Damien needed this while he was on Mars (in the movie "The Martian")!

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	*[NULL]*	32	20	*[SPACE]*	64	40	**@**	96	60	**`**
1	1	*[START OF HEADING]*	33	21	**!**	65	41	**A**	97	61	**a**
2	2	*[START OF TEXT]*	34	22	**"**	66	42	**B**	98	62	**b**
3	3	*[END OF TEXT]*	35	23	**#**	67	43	**C**	99	63	**c**
4	4	*[END OF TRANSMISSION]*	36	24	**$**	68	44	**D**	100	64	**d**
5	5	*[ENQUIRY]*	37	25	**%**	69	45	**E**	101	65	**e**
6	6	*[ACKNOWLEDGE]*	38	26	**&**	70	46	**F**	102	66	**f**
7	7	*[BELL]*	39	27	**'**	71	47	**G**	103	67	**g**
8	8	*[BACKSPACE]*	40	28	**(**	72	48	**H**	104	68	**h**
9	9	*[HORIZONTAL TAB]*	41	29	**)**	73	49	**I**	105	69	**i**
10	A	*[LINE FEED]*	42	2A	*****	74	4A	**J**	106	6A	**j**
11	B	*[VERTICAL TAB]*	43	2B	**+**	75	4B	**K**	107	6B	**k**
12	C	*[FORM FEED]*	44	2C	**,**	76	4C	**L**	108	6C	**l**
13	D	*[CARRIAGE RETURN]*	45	2D	**-**	77	4D	**M**	109	6D	**m**
14	E	*[SHIFT OUT]*	46	2E	**.**	78	4E	**N**	110	6E	**n**
15	F	*[SHIFT IN]*	47	2F	**/**	79	4F	**O**	111	6F	**o**
16	10	*[DATA LINK ESCAPE]*	48	30	**0**	80	50	**P**	112	70	**p**
17	11	*[DEVICE CONTROL 1]*	49	31	**1**	81	51	**Q**	113	71	**q**
18	12	*[DEVICE CONTROL 2]*	50	32	**2**	82	52	**R**	114	72	**r**
19	13	*[DEVICE CONTROL 3]*	51	33	**3**	83	53	**S**	115	73	**s**
20	14	*[DEVICE CONTROL 4]*	52	34	**4**	84	54	**T**	116	74	**t**
21	15	*[NEGATIVE ACKNOWLEDGE]*	53	35	**5**	85	55	**U**	117	75	**u**
22	16	*[SYNCHRONOUS IDLE]*	54	36	**6**	86	56	**V**	118	76	**v**
23	17	*[ENG OF TRANS. BLOCK]*	55	37	**7**	87	57	**W**	119	77	**w**
24	18	*[CANCEL]*	56	38	**8**	88	58	**X**	120	78	**x**
25	19	*[END OF MEDIUM]*	57	39	**9**	89	59	**Y**	121	79	**y**
26	1A	*[SUBSTITUTE]*	58	3A	**:**	90	5A	**Z**	122	7A	**z**
27	1B	*[ESCAPE]*	59	3B	**;**	91	5B	**[**	123	7B	**{**
28	1C	*[FILE SEPARATOR]*	60	3C	**<**	92	5C	****	124	7C	**|**
29	1D	*[GROUP SEPARATOR]*	61	3D	**=**	93	5D	**]**	125	7D	**}**
30	1E	*[RECORD SEPARATOR]*	62	3E	**>**	94	5E	**^**	126	7E	**~**
31	1F	*[UNIT SEPARATOR]*	63	3F	**?**	95	5F	**_**	127	7F	*[DEL]*