

converting string to hex

system 

Feb '19 post #2

Im trying to convert a string like hello world to hex

That makes NO sense. Hex is one way of showing a numeric value. The value 10, in a base 10 system would be represented as 0xA in Hex.

Does anyone know how I can fix this?

I can't even understand what is broken. What you are doing, even if you were doing it properly, makes not sense. And, you are NOT doing it properly.

```
Serial.print("%x" + str[i]);
```

Adding a string and a character does not make sense. What does your calculator show when you try to add %x and h? How do you even enter those into the calculator?

The + operator does addition. Nothing else, unless there are class instances involved, and the class overloads the + operator.

[Skip to main content](#) no class instances involved, so no class overloaded the + operator.

Iesept

Feb '19 post #3

convert a string like hello world to hex and get the amount of bytes

The amount of bytes in a string is just the number of characters, strlen() will give you that number

If you would want to know the sum of all bytes in the string, do sth like:

```
int sum = 0;
for (int i = 0; i < strlen(str); i++) sum += (str[i] - '0');
Serial.println (sum);
```

Not tested, but it should give you the sum of all bytes in the string.

GolamMostafa Karma: 500+

Feb '19 post #4

@OP

1. You have this string of characters: **Hello World** consisting of 11 characters (including the space between Hello and World).

2. If you replace every character of the string of Step-1 by its corresponding ASCII Code (Fig-1), you will come up like this: 48 65 6C 6C 6F 20 57 6F 72 6C 64; where, 48 is in hex form for the ASCII Codes (01001000) of **H**; 65 is the hex form for the ASCII Code of **e**, and so on. There are 11 bytes of hex formatted data for the 11 characters of the 'Hello World' string

					extra bit for the second set of characters								
b7 →					0	0	0	0	1	1	1	1	
b6 →					0	0	0	1	1	0	0	1	1
b5 →					0	1	0	1	0	1	0	1	
b4 →													
r o w	b3	b2	b1	b0	r\s	0	1	2	3	4	5	6	7
						c o l u m n							
	0	0	0	0	0	NUL	DLE	SP	0	@	P	`	p
	0	0	0	1	1	SOH	DC1	!	1	A	Q	a	q
	0	0	1	0	2	STX	DC2	"	2	B	R	q	r
	0	0	1	1	3	ETX	DC3	#	3	C	S	c	s
	0	1	0	0	4	EOT	DC4	\$	4	D	T	d	t
	0	1	0	1	5	ENQ	NAK	%	5	E	U	e	u
	0	1	1	0	6	ACK	SYN	&	6	F	U	f	u
	0	1	1	1	7	BEL	ETB	'	7	G	W	g	w
	1	0	0	0	8	BS	CAN	(8	H	X	h	x
	1	0	0	1	9	HT	EM)	9	I	T	i	t
	1	0	1	0	A	LF	SUB	*	:	J	Z	j	z
	1	0	1	1	B	UT	ESC	+	;	K	[k	{
	1	1	0	0	C	FF	FS	,	<	L	\	l	
	1	1	0	1	D	CR	GS	-	=	M]	m	}
1	1	1	0	E	SO	RS	.	>	N	^	n	~	
1	1	1	1	F	SI	US	/	?	O	_	o	DEL	

Figure-1: ASCII Code Chart for the characters of English Language Alphabet and Control Characters

3. When you declare this array: `char msgArray[] = "Hello World";`, the array locations are automatically filled up by the ASCII Codes of the characters of the 'Hello World' string. This is to say that location msgArray[0] is filled up by the ASCII Code (01001000 = 0x48) of **H**, location msgArray[1] is filled up by the ASCII Code (01100101 = 0x65) of **e**, and so on. At the end, one more location (location msgArray[11]) will be filled up by the ASCII Code (00000000 = 0x00) of the so called NULL Byte.

4. If you want to see the print out of the ASCII Codes of the 'Hello World' string in hex format in the Serial Monitor, you just execute the following sketch. The sketch also prints out the length of the string (11).

```
char msgArray[] = "Hello World";
```

[Skip to main content](#)

```

Serial.begin(9600);

Serial.println(msgArray);
for (int i = 0; i<sizeof(msgArray)-1; i++)
{
    Serial.print(msgArray[i], HEX);//excludes NULL byte
}
Serial.println();
Serial.println(sizeof(msgArray)-1, DEC);//excluding NULL byte

}

void loop()
{
}

```

					extra bit for the second set of characters								
b7 →					0 0 0 0 1 1 1 1								
b6 →					0 0 1 1 0 0 1 1								
b5 →					0 0 1 1 0 0 1 1								
b4 →					0 1 0 1 0 1 0 1								
r o w	b3 b2 b1 b0				r\s	0	1	2	3	4	5	6	7
						c o l u m n							
	0	0	0	0	0	NUL	DLE	SP	0	@	P	`	p
	0	0	0	1	1	SOH	DC1	!	1	A	Q	a	q
	0	0	1	0	2	STX	DC2	"	2	B	R	q	r
	0	0	1	1	3	ETX	DC3	#	3	C	S	c	s
	0	1	0	0	4	EOT	DC4	\$	4	D	T	d	t
	0	1	0	1	5	ENQ	NAK	%	5	E	U	e	u
	0	1	1	0	6	ACK	SYN	&	6	F	U	f	u
	0	1	1	1	7	BEL	ETB	'	7	G	W	g	w
	1	0	0	0	8	BS	CAN	(8	H	X	h	x
	1	0	0	1	9	HT	EM)	9	I	T	i	t
	1	0	1	0	A	LF	SUB	*	:	J	Z	j	z
	1	0	1	1	B	UT	ESC	+	;	K	[k	{
	1	1	0	0	C	FF	FS	,	<	L	\	l	
	1	1	0	1	D	CR	GS	-	=	M]	m	}
	1	1	1	0	E	SO	RS	.	>	N	^	n	~
1	1	1	1	F	SI	US	/	?	O	_	o	DEL	

hoekynl

Feb '19 post #5

GolamMostafa:
@OP

1. You have this string of characters: **Hello World** consisting of 11 characters (including the space between Hello and World).

2. If you replace every character of the string of Step-1 by its corresponding ASCII Code (Fig-1), you will come up like this: 48 65 6C 6C 6F 20 57 6F 72 6C 64; where, 48 is in hex form for the ASCII Codes (01001000) of **H**; 65 is the hex form for the ASCII Code of **o**, and so on. There are 11 bytes of hex formatted data for the 11 characters of the 'Hello World' string

[Skip to main content](#)

<div><div><div>b7</div><div>b6</div><div>b5</div><div>b4</div></div><div><div>→</div><div>→</div><div>→</div><div>→</div></div></div>					<div><div>extra bit for the second set of characters</div><div><div>0</div><div>0</div><div>0</div><div>0</div><div>1</div><div>1</div><div>1</div><div>1</div><div>1</div><div>1</div><div>1</div><div>1</div></div><div><div>0</div><div>0</div><div>0</div><div>1</div><div>1</div><div>0</div><div>0</div><div>1</div><div>1</div><div>1</div><div>1</div><div>1</div></div><div><div>0</div><div>1</div><div>0</div><div>1</div><div>0</div><div>1</div><div>0</div><div>1</div><div>0</div><div>1</div><div>0</div><div>1</div></div></div>												
					<div><div><div>b3 b2 b1 b0</div><div>r\s</div></div><div><div>0</div><div>1</div><div>2</div><div>3</div><div>4</div><div>5</div><div>6</div><div>7</div></div></div> <div><div>c o l u m n</div></div>												
<div><div>r o w</div></div>					<div>0</div> <div>0</div> <div>0</div> <div>0</div>				0	NUL	DLE	SP	0	@	P	`	p
					<div>0</div> <div>0</div> <div>0</div> <div>1</div>				1	SOH	DC1	!	1	A	Q	a	q
					<div>0</div> <div>0</div> <div>1</div> <div>0</div>				2	STX	DC2	"	2	B	R	q	r
					<div>0</div> <div>0</div> <div>1</div> <div>1</div>				3	ETX	DC3	#	3	C	S	c	s
					<div>0</div> <div>1</div> <div>0</div> <div>0</div>				4	EOT	DC4	\$	4	D	T	d	t
					<div>0</div> <div>1</div> <div>0</div> <div>1</div>				5	ENQ	NAK	%	5	E	U	e	u
					<div>0</div> <div>1</div> <div>1</div> <div>0</div>				6	ACK	SYN	&	6	F	U	f	u
					<div>0</div> <div>1</div> <div>1</div> <div>1</div>				7	BEL	ETB	'	7	G	W	g	w
					<div>1</div> <div>0</div> <div>0</div> <div>0</div>				8	BS	CAN	(8	H	X	h	x
					<div>1</div> <div>0</div> <div>0</div> <div>1</div>				9	HT	EM)	9	I	T	i	t
					<div>1</div> <div>0</div> <div>1</div> <div>0</div>				A	LF	SUB	*	:	J	Z	j	z
					<div>1</div> <div>0</div> <div>1</div> <div>1</div>				B	UT	ESC	+	;	K	[k	{
					<div>1</div> <div>1</div> <div>0</div> <div>0</div>				C	FF	FS	,	<	L	\	l	
<div>1</div> <div>1</div> <div>0</div> <div>1</div>				D	CR	GS	-	=	M]	m	}					
<div>1</div> <div>1</div> <div>1</div> <div>0</div>				E	SO	RS	.	>	N	^	n	~					
<div>1</div> <div>1</div> <div>1</div> <div>1</div>				F	SI	US	/	?	O	_	o	DEL					

Figure-1: ASCII Code Chart for the characters of English Language Alphabet and Control Characters

3. When you declare this array: **char msgArray[] = "Hello World";**, the array locations are automatically filled up by the ASCII Codes of the characters of the 'Hello World' string. This is to say that location msgArray[0] is filled up by the ASCII Code (01001000 = 0x48) of **H**, location msgArray[1] is filled up by the ASCII Code (01100101 = 0x65) of **e**, and so on. At the end, one more location (location msgArray[11]) will be filled up by the ASCII Code (00000000 = 0x00) of the so called NULL Byte.

4. If you want to see the print out of the ASCII Codes of the 'Hello World' string in hex format in the Serial Monitor, you just execute the following sketch. The sketch also prints out the length of the string (11).

```
char msgArray[] = "Hello World";
```

```
void setup()
{
  Serial.begin(9600);

  Serial.println(msgArray);
  for (int i = 0; i<sizeof(msgArray)-1; i++)
  {
    Serial.print(msgArray[i], HEX);//excludes NULL byte
  }
  Serial.println();
  Serial.println(sizeof(msgArray)-1, DEC);//excluding NULL byte
}

void loop()
{
}
```

Thanks, this seems to work but how do I make the HEX and Byte parts into variables I can use? That is the main part im stuck on at this moment

GolamMostafa Karma: 500+

Feb '19 post #6

[Skip to main content](#)

[Skip to main content](#)

can use? That is the main part im stuck on at this moment

That means that you want pack/insert the parameters (11 and 48656C6C6F20576F726C64) into the indicated/relevant places of the following structure:

```
AT+NSOST=0,"172.27.131.100",15683,11,"48656c6c6f20576f726c64"
```

If you kindly explain what the above thing is, we may try to help you. Is it a data structure like?:

```
struct myData
{
    int x;//,
    char data1Array[50] = "";
    int y;//,
    int z;
    char data2Array[50] = "";
};
```

hoekynl

Feb '19 post #7

GolamMostafa:

That means that you want pack/insert the parameters (11 and 48656C6C6F20576F726C64) into the indicated/relevant places of the following structure:

```
AT+NSOST=0,"172.27.131.100",15683,11,"48656c6c6f20576f726c64"
```

If you kindly explain what the above thing is, we may try to help you. Is it a data structure like?:

```
struct myData
```

```
{
    int x,
    char data1Array[50];
    int y,
    int z;
    char data2Array[50];
};
```

It is an example of the nb-iot from t-mobile: [Internet of Things voor je bedrijf | T-Mobile Ondernemen](#)

In the [documentation for the BC95 chip](#) im using the AT+NSOST is structured like this:

```
AT+NSOST=<remote_addr>,<remote_port>,,
```

Parameter

Socket number returned by AT+NSOCR (always 0)

<remote_addr> IPv4 A dot notation IP address (always 172.27.131.100)

<remote_port> A number in the range 0-65535. This is the remote port on which messages will be received (always 15683)

Decimal length of data to be sent

Data received in hex string format

Eventually my arduino (seeeduino lotus) would get temperature, humidity, db levels and weight from a beehive and transfer it via nb-iot so I can monitor it remotely

So what I want to do is insert my information into the function which converts it to HEX and the amount of bytes and then pass it to the command to be sent

[Skip to main content](#) mean you needed 😊

GolamMostafa Karma: 500+
Feb '19 **post #8**

@OP

Check if the following codes could extend some kinds of help to you ?

```
//AT+NSOST=0, "172.27.131.100", 15683, 11, "48656c6c6f20576f726c64"
char msgArray[] = "Hello World";
```

```
struct
{
    int x;
    char *data1Array;
    int y;
    int z;
    char *data2Array;
} myVar;

void setup()
{
    Serial.begin(9600);
    myVar.x = 0;
    myVar.data1Array = "172.27.131.100";
    myVar.y = 15683;
    myVar.z = sizeof(msgArray) - 1;
    myVar.data2Array = msgArray;
    Serial.println(myVar.data1Array);
}

void loop()
{
}
}
```

hoekynl
Feb '19 **post #9**

GolamMostafa:
@OP

Check if the following codes could extend some kinds of help to you ?

```
//AT+NSOST=0, "172.27.131.100", 15683, 11, "48656c6c6f20576f726c64"
```

```
char msgArray[] = "Hello World";
```

```
struct
{
    int x;
    char *data1Array;
    int y;
    int z;
    char *data2Array;
} myVar;

void setup()
{
    Serial.begin(9600);
    myVar.x = 0;
    myVar.data1Array = "172.27.131.100";
    myVar.y = 15683;
    myVar.z = sizeof(msgArray) - 1;
Skip to main content sgArray;
```

```

Serial.println(myVar.data1Array);
}

void loop()
{
}

```

This would be a nice way to do this but myVar.data2Array is not in HEX so it would not work



system

Feb '19 post #10

hoekynl:

This would be a nice way to do this but myVar.data2Array is not in HEX so it would not work



If the struct, and the items in the struct, are useful, then it is trivial to add a method to the struct to convert the string to the representation that you want.

GolamMostafa Karma: 500+

Feb '19 post #11

hoekynl:

This would be a nice way to do this but myVar.data2Array is not in HEX so it would not work



PaulS:

If the struct, and the items in the struct, are useful, then it is trivial to add a method to the struct to convert the string to the representation that you want.

The variable myVar.data2Array (an array) contains the information 'Hello World' (in the high level perception). In the actual low level, the array (data2Array) contains in its every location the 8-bit ASCII Codes (01001000 = 0x48 for H and so on.) for the characters of the message (Hello World). This can be easily (in mathematical language as @PaulS has mentioned – it is trivial) verified by executing the following code:

```

Serial.println(myVar.data2Array[0], BIN); //shows: 01001000
Serial.println(myVar.data2Array[0], HEX); //shows: 48
Serial.println((char)myVar.dataArray);    //shows: H

```

Closed on May 6, '21

[Back to top](#)

[Help Center](#)

[Contact Us](#)

[Trademark & Copyright](#)

[Brand Guidelines](#)

[Distributors](#)

[Careers](#)

FOLLOW US