5.3 ASCII DATA REPRESENTATION

Successful communication requires standardized data representation so that people and computers around the world can share the same information. Alphanumeric characters are represented by a seven-bit standard representation known as the *American Standard Code for Information Interchange*, or ASCII. ASCII also includes punctuation marks and invisible control codes used to help in the display and transfer of data. ASCII was first published in 1968 by the *American National Standards Institute*, or ANSI. The original ASCII standard lacked provisions for many commonly used grammatical symbols in languages other than English. Since 1968, there have been many extensions to ASCII that have varying support throughout the world according to the prevalent language in each country. In the United States, an eight-bit ASCII variant is commonly supported that adds graphical symbols and some of the more common foreign language punctuation symbols. The original seven-bit ANSI standard ASCII mapping is shown in Table 5.1. The mappings below 0x20 are invisible control codes such as tab (0x09), carriage return (0x0D), and line-feed (0x0A). Some of the control codes are not in widespread use anymore.

5.4 RS-232

Aside from a common data representation format, communication signaling such as framing or error detection also requires standardization so that equipment manufactured by different companies can exchange information. When one begins discussing communications, an unstoppable journey into the sometimes mysterious world of industry standards begins. Navigating these standards can be tricky because of subtle differences in terminology between related standards and the everyday jargon to which the engineering community has grown accustomed. Standards are living documents that are periodically updated, revised, or replaced. This shifting base of documentation can add other challenges to fully complying with a standard.

One of the most ubiquitous serial communications schemes in use is defined by the *RS-232* family of standards. Most UARTs are designed specifically to support RS-232. Standards purists may balk at the common reference to RS-232 in the modern context, for several reasons. First, the original RS-232 document has long since been superseded by multiple revisions. Second, its name was changed first to EIA-232, then to EIA/TIA-232. And third, RS-232 is but one of a set of related standards that address asynchronous serial communications. These standards have been developed under the auspices of the Electronics Industry Alliance (formerly the Electronics Industry Association) and Telecommunications Industry Association. Technically, EIA/TIA-232 (first introduced in 1962 as RS-232) standardizes the 25-pin D-subminiature (DB25) connector and pin assignment along with an obsolete electrical specification that had limited range. EIA/TIA-423 standardizes the modern electrical characteristics that enable communication at speeds up to 100 kbps over short distances (10 m). EIA/TIA-574 standardizes the popular nine-pin DE9 connector that is used on most new "RS-232" equipped devices. These days, when most people talk about an RS-232 port, they are referring to the overall RS-232 family of related serial interfaces. In fairness to standards purists, this loose terminology is partially responsible for confusion among those who implement and use RS-232. From a practical perspective, however, it is most common to use the term RS-232 with additional qualifiers (e.g., 9-pin or 25-pin) to convey your point. In fact, if you start mentioning EIA/TIA-574 and 423, you will probably be met by blank stares from most engineers. This somewhat shady practice is continued here because of its widespread acceptance in industry.

RS-232 specifies that the least-significant bit of a byte is transmitted first and is framed by a single start bit and one or two stop bits. Common RS-232 data rates are known to many computer users.

TABLE 5.1 Seven-bit ASCII Character Mapping

Decimal	Hex	Value	Decimal	Hex	Value	Decimal	Hex	Value	Decimal	Hex	Value
0	0x00	NUL	32	0x20	SP	64	0x40	@	96	0x60	`
1	0x01	SOH	33	0x21	!	65	0x41	A	97	0x61	a
2	0x02	STX	34	0x22	"	66	0x42	B	98	0x62	b
3	0x03	ETX	35	0x23	#	67	0x43	C	99	0x63	c
4	0x04	EOT	36	0x24	$	68	0x44	D	100	0x64	d
5	0x05	ENQ	37	0x25	%	69	0x45	E	101	0x65	e
6	0x06	ACK	38	0x26	&	70	0x46	F	102	0x66	f
7	0x07	BEL	39	0x27	'	71	0x47	G	103	0x67	g
8	0x08	BS	40	0x28	(	72	0x48	H	104	0x68	h
9	0x09	HT	41	0x29	)	73	0x49	I	105	0x69	i
10	0x0A	LF	42	0x2A	*	74	0x4A	J	106	0x6A	j
11	0x0B	VT	43	0x2B	+	75	0x4B	K	107	0x6B	k
12	0x0C	FF	44	0x2C	,	76	0x4C	L	108	0x6C	l
13	0x0D	CR	45	0x2D	-	77	0x4D	M	109	0x6D	m
14	0x0E	SO	46	0x2E	.	78	0x4E	N	110	0x6E	n
15	0x0F	SI	47	0x2F	/	79	0x4F	O	111	0x6F	o
16	0x10	DLE	48	0x30	0	80	0x50	P	112	0x70	p
17	0x11	DC1/XON	49	0x31	1	81	0x51	Q	113	0x71	q
18	0x12	DC2	50	0x32	2	82	0x52	R	114	0x72	r
19	0x13	DC3/XOFF	51	0x33	3	83	0x53	S	115	0x73	s
20	0x14	DC4	52	0x34	4	84	0x54	T	116	0x74	t
21	0x15	NAK	53	0x35	5	85	0x55	U	117	0x75	u
22	0x16	SYN	54	0x36	6	86	0x56	V	118	0x76	v
23	0x17	ETB	55	0x37	7	87	0x57	W	119	0x77	w
24	0x18	CAN	56	0x38	8	88	0x58	X	120	0x78	x
25	0x19	EM	57	0x39	9	89	0x59	Y	121	0x79	y
26	0x1A	SUB	58	0x3A	:	90	0x5A	Z	122	0x7A	z
27	0x1B	ESC	59	0x3B	;	91	0x5B	[	123	0x7B	{
28	0x1C	FS	60	0x3C	<	92	0x5C	\	124	0x7C	|
29	0x1D	GS	61	0x3D	=	93	0x5D	]	125	0x7D	}
30	0x1E	RS	62	0x3E	>	94	0x5E	^	126	0x7E	~
31	0x1F	US	63	0x3F	?	95	0x5F	_	127	0x7F	DEL