RS232Cテストデータ送信ユニット

概要

PICを使った電子工作では、RS232Cを使用する場合が多々あります。 そして、その動作確認を実施する際には、次のようなことを確認する必要があります。

- 正常にデータの送信/受信をしているかを確認する。
- 連続的にデータの送信/受信をさせ耐久性を確認する。
- 高負荷のデータの送信/受信をさせ性能を確認する。

※以前に製作したLCDモニターは、RS232Cのデータを受信してLCDに表示させることが出来るので、製作したPICからデータが正常に送信されているかどうかの判断に使用することが出来ます。

今回は、次の4種類のテストデータを送信するユニットを製作しました。

- 0123456789
- ABCDEFGHIJKLMNOPQRSTUVWXYZ
- abcdefghijklmnopqrstuvwxyz
- 0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz

また、連続的にデータを送信するか、送信毎に、遅延(1秒)を入れるかを設定することが出来るようにしました。

動作原理

送信するデータは、ASCIIコードです。ASCIIコードは、7ビットで構成されており、今回の送信データは、その内 の英数文字を送信できるようにしました。(通信速度は、9600bps固定)

<数字文字> '0'~'9'は、16進数の0x30~0x39になります。

<英大文字> 'A'~'Z'は、16進数の0x41~0x5Aになります。

<英小文字> 'a'~'z'は、16進数の0x61~0x7Aになります。

<ASCIIコード表>

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

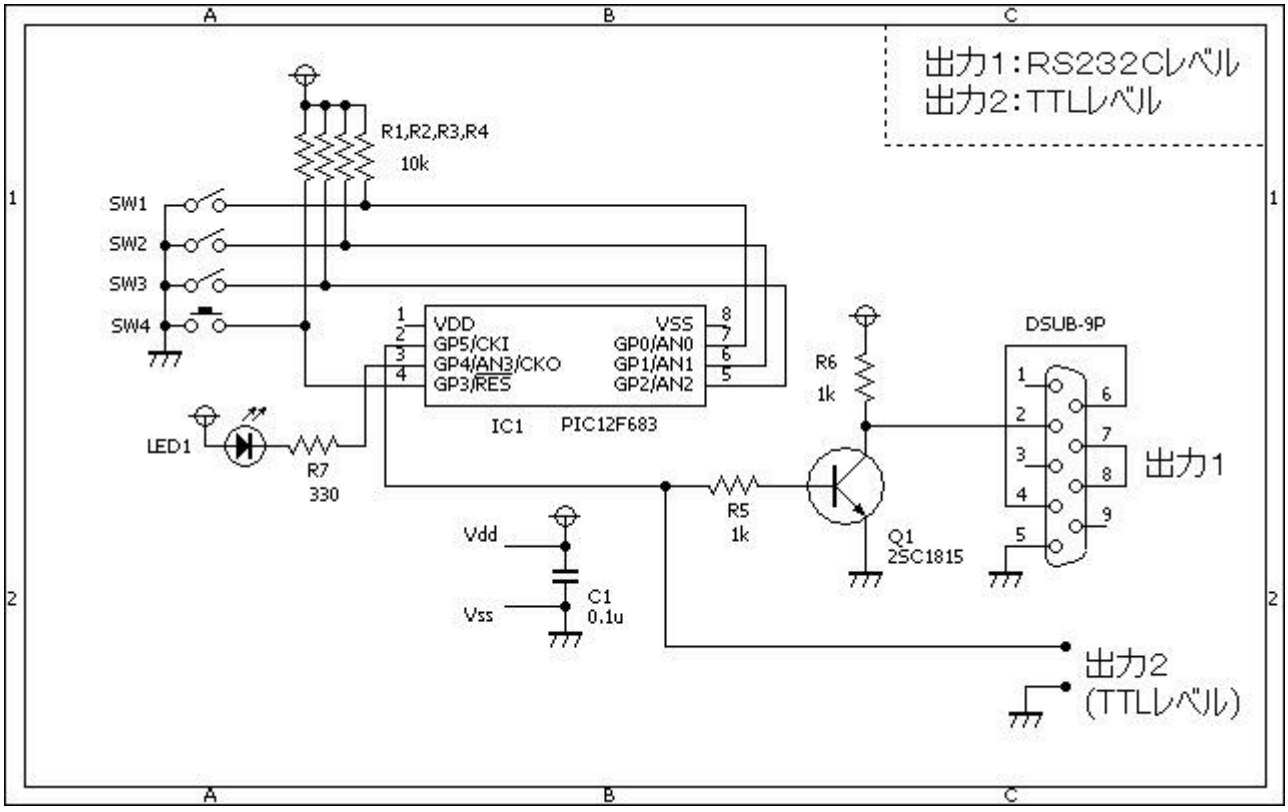
<送信データのパターン>

- 0123456789
  ※SW1=OFF、SW2=OFF
- ABCDEFGHIJKLMNOPQRSTUVWXYZ
  ※SW1=ON、SW2=OFF
- abcdefghijklmnopqrstuvwxyz
  ※SW1=OFF、SW2=ON
- 0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz
  ※SW1=ON、SW2=ON

<開始と停止> プッシュスイッチ(SW4)を、押下する毎に開始と停止を繰り返します。開始中はLED1が点灯します。

<遅延> 上記のパターンデータを送信する毎に、1秒の遅延をするかどうかを指定します。 ※遅延しない、SW3=OFF ※遅延する、SW3=ON

回路図



ソースコード

Rs232cTestV3.c

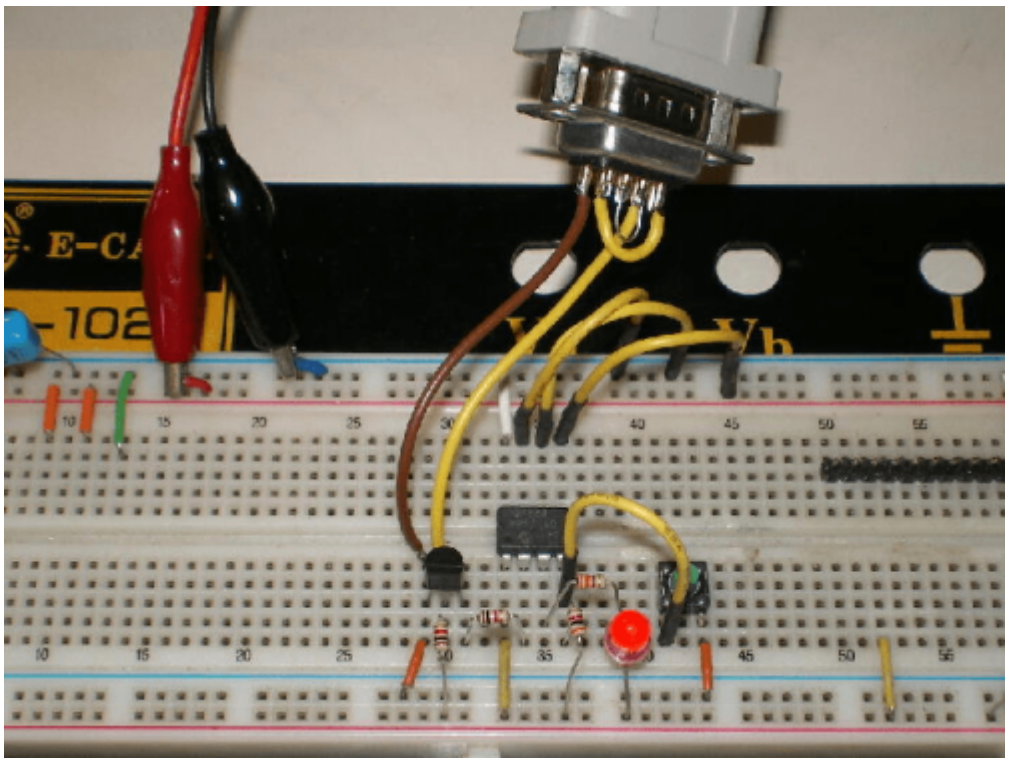
```
//*****************************************************************************
*
/*
□□□RS232Cテストデータ送信用」
  ■通信速度
□□□□□□□□□□固定
  ■送信パターン
□□□"0123456789"
□□□"ABCDEFGHIJKLMNOPQRSTUVWXYZ"
□□□"abcdefghijklmnopqrstuvwxyz"
    ・上記３つを合わせたパターン
*/
//*****************************************************************************
*

#define       SW1             GPIO.F0
#define       SW2             GPIO.F1
#define       SW3             GPIO.F2
#define       SW4             GPIO.F3

#define       LED             GPIO.F4

#define       CR              0x0D
#define       LF              0x0A
```

```
#define       ON            0
#define       OFF            1

//*********************************************************************
*

void   dataSend1()
{
    char    cnt;
    //
    for (cnt = '0'; cnt <= '9'; cnt++) {
        Soft_Uart_Write(cnt);
    }
}

//*********************************************************************
*

void   dataSend2()
{
    char    cnt;
    //
    for (cnt = 'A'; cnt <= 'Z'; cnt++) {
        Soft_Uart_Write(cnt);
    }
}

//*********************************************************************
*

void   dataSend3()
{
    char    cnt;
    //
    for (cnt = 'a'; cnt <= 'z'; cnt++) {
        Soft_Uart_Write(cnt);
    }
}

//*********************************************************************
*

void main()
{
    char    cnt;
    //
    OSCCON = 0b01110000;     //クロックは8Mhz
    ANSEL  = 0b00000000;     //今回は使用しない。
    CMCON0 = 0b00000111;     //今回は使用しない。
```

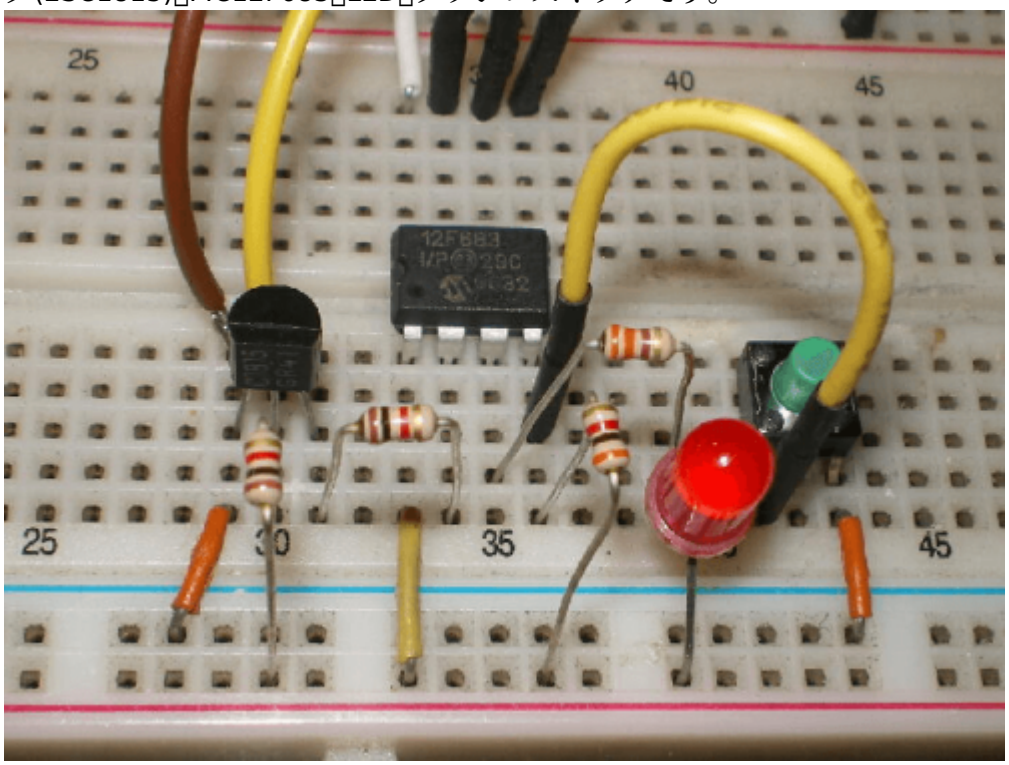
```
    TRISIO = 0b00001111;
    //USARTの初期化
    Soft_Uart_Init(GPIO, 3, 5, 9600, 0);
    for (cnt = 0; cnt < 5; cnt++) {
        LED = ON;
        Delay_ms(100);
        LED = OFF;
        Delay_ms(100);
    }
    //
    while (1) {
        //開始スイッチが押されるのを待つ。
        while (Button(&GPIO, 3, 1, 0) == 0)
            ;
        while (Button(&GPIO, 3, 1, 1) == 0)
            ;
        //
        LED = ON;
        while (1) {
            if ((SW1 == OFF) && (SW2 == OFF)) {
                dataSend1();
            }
            if ((SW1 == ON) && (SW2 == OFF)) {
                dataSend2();
            }
            if ((SW1 == OFF) && (SW2 == ON)) {
                dataSend3();
            }
            if ((SW1 == ON) && (SW2 == ON)) {
                dataSend1();
                dataSend2();
                dataSend3();
            }
            Soft_Uart_Write(CR);
            Soft_Uart_Write(LF);
            //ディレイ（遅延）の判断
            if (SW3 == OFF) {
                Delay_ms(1000);
            }
            //停止スイッチが押されたかを確認。
            if (SW4 == ON) {
                while (Button(&GPIO, 3, 1, 0) == 0)
                    ;
                while (Button(&GPIO, 3, 1, 1) == 0)
                    ;
                break;
            }
        }
        LED = OFF;
    }
}
```

```
//*************************************************************************
*
```

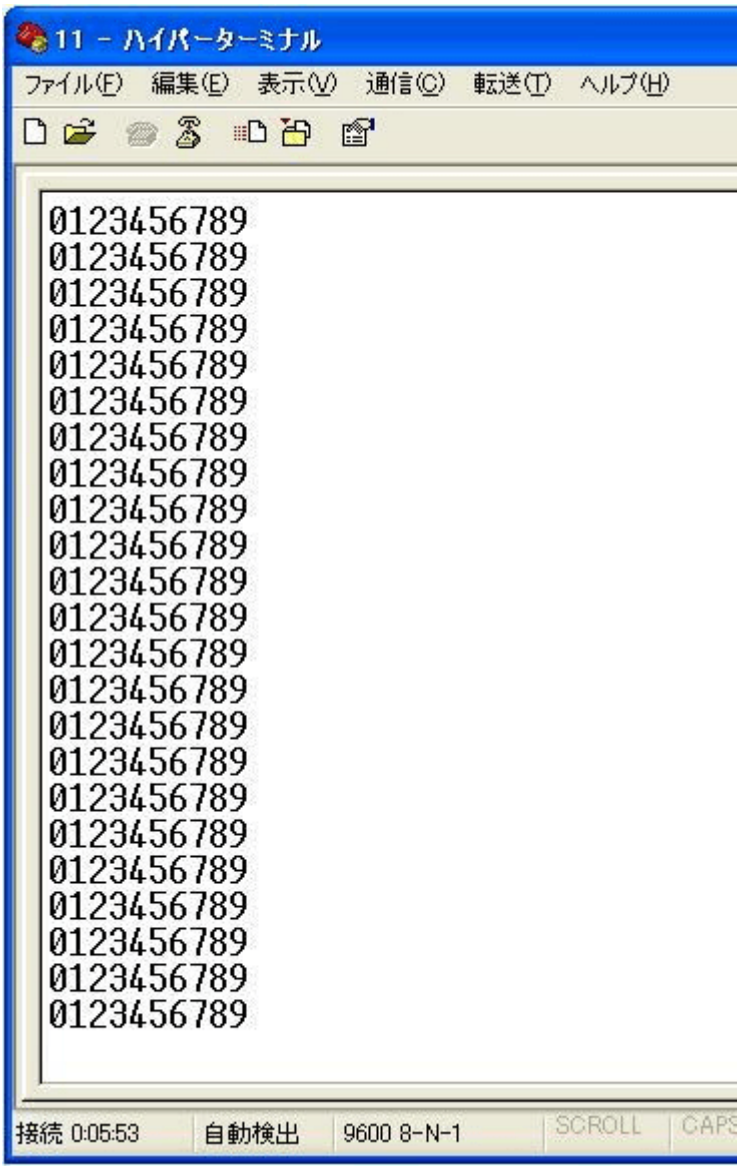
動作確認



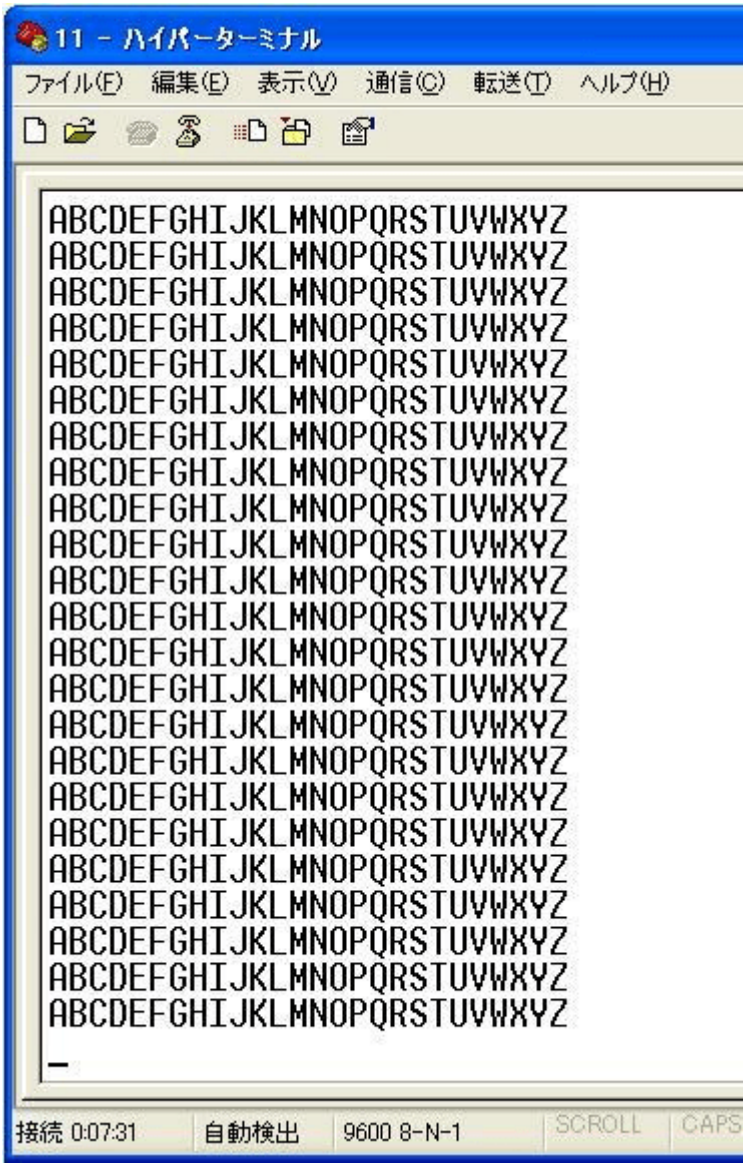
左側から、トランジスタ(2SC1815)、PIC12F683、LED、プッシュスイッチです。



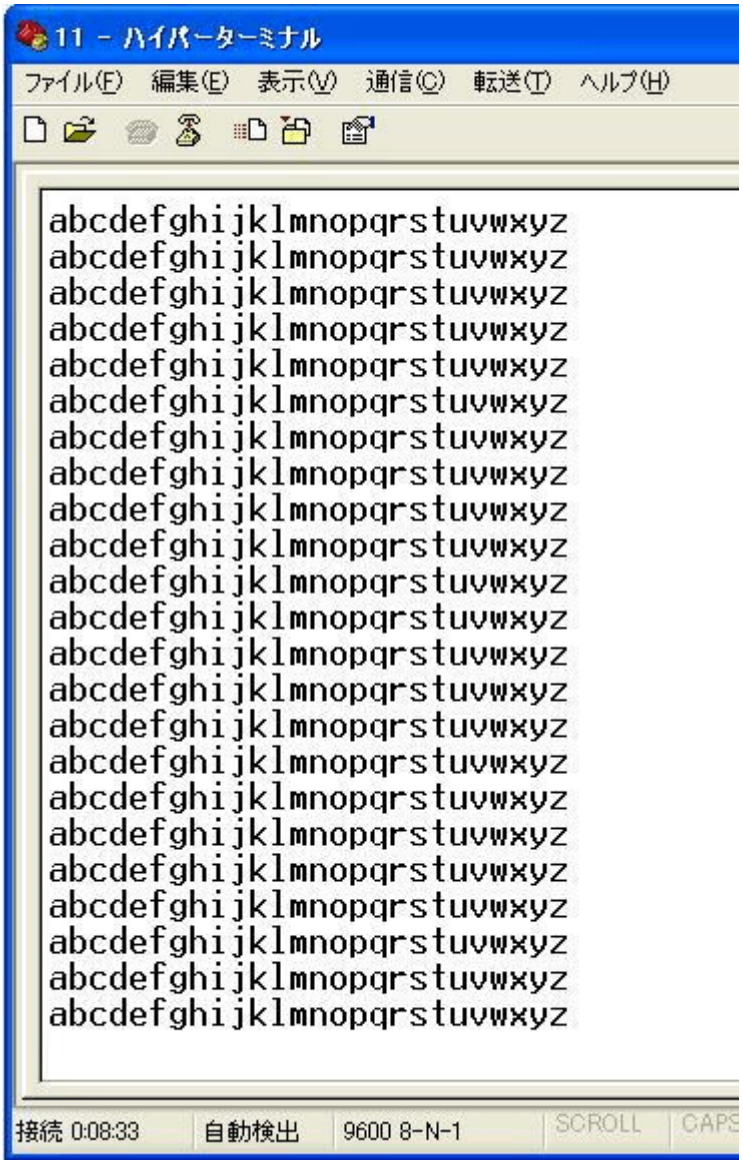
数字文字の送信パターンで



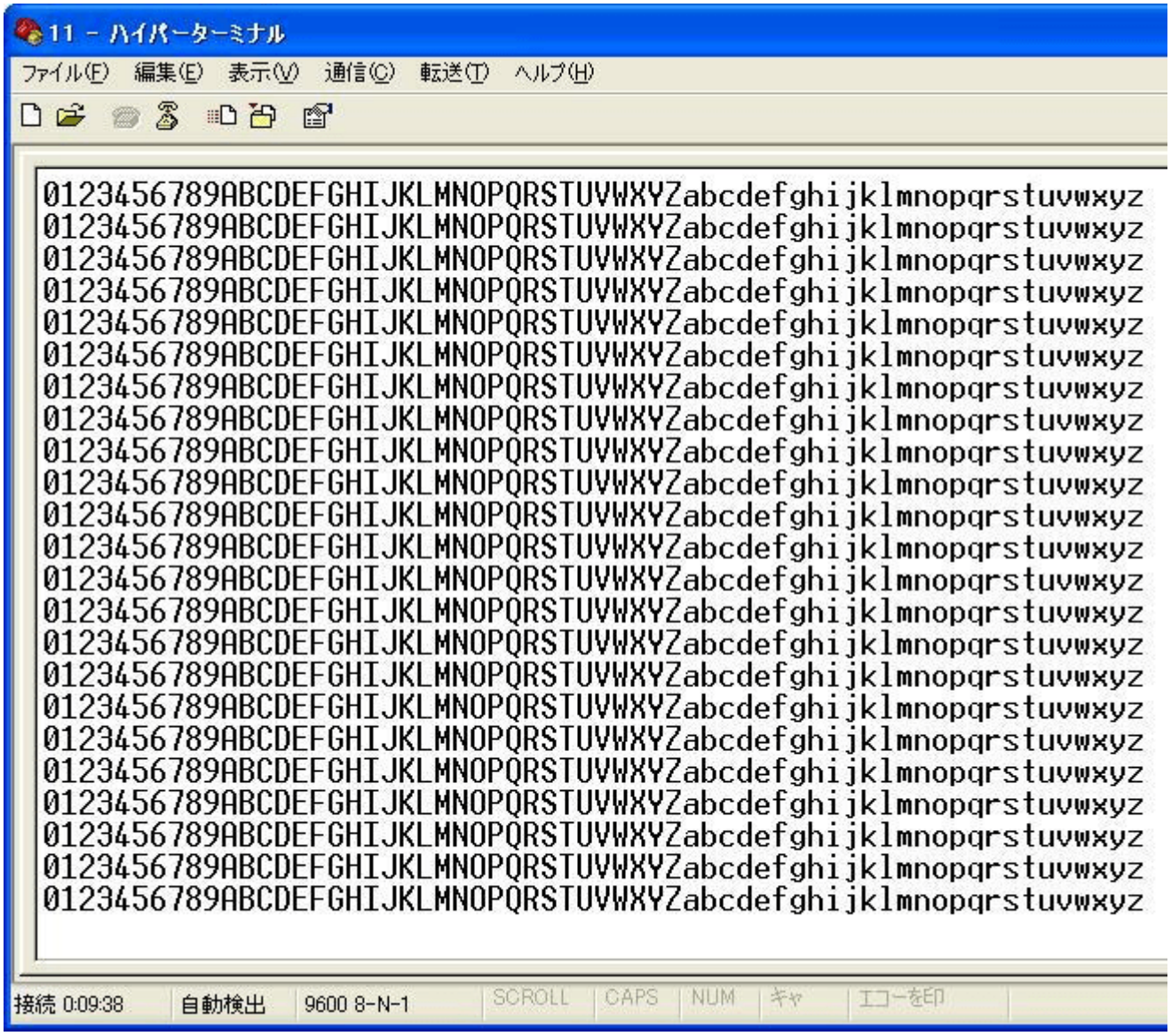
す。英大文字の送信パターンです。



英小文字の送信パターンです。



数字文字+英大文字+英小文字の送信**パターンです**。



如何ですか? 本ユニットを一台手元に用意しておくとRS232Cを使用する電子工作には、とても重宝しますよ!{ }!