

Re: It is possible for 89C51 to receive 7-bit RS232 data?

## Re: It is possible for 89C51 to receive 7-bit RS232 data?

---

*Source:* <http://coding.derkeiler.com/Archive/General/comp.arch.embedded/2008-11/msg00418.html>

---

- *From:* "jusmansulaiman" <jusmansulaiman@xxxxxxxx>
  - *Date:* Tue, 18 Nov 2008 18:51:42 -0600
- 

Jim Flanagan wrote:

So, you would set up your 8051 to 8 bit mode, which it appears you

do.

You would receive the data and 'strip' off the MSB (and with 7fh).

Leaving the 7bit ascii char for you to process.

```
Yes, to the lines :
FetchCharacter:
jnb RI,FetchCharacter
clr RI
mov A,SBUF
ret
I added
anl A,#7Fh
```

before the ret,

so as to strip off the parity bit appeared at 7th bit at the data. Then

I

```
subtracted it with 30h and used it as a pointer to a lookup table.
call FetchCharacter
```

YOU NEED TO CLEAR THE CARRY HERE AS THE SUBB INSTRUCTION IS - SUBTRACT WITH BORROW.

Re: It is possible for 89C51 to receive 7-bit RS232 data?

>>> clr c <<<<

```
subb A,#30h
mov DPTR,#Table1
movc A,@A+DPTR
```

.  
.

But it didnt work.

Valid data are prefixed by the signature ST, and first I took only the character S for the program to process the 6 byte numeric data began at

7th

byte. When I left the S unstripped, the program acknowledge the S

character

and went to the 7th byte. Later, I realized that the S character had

parity

bit "0" so it is the same whether you strip it or not.

I found that the "ANL A,#7Fh" could not work at all. For the numeric

data

I tried with :

```
ANL A,#00001111b
```

If this worked and the subb didn't that tells me the 'carry' not being cleared probably caused your subb to not work correctly

and delete the useless sub A,#30h. It worked OK !!

You mean subb?

Later I realized that I had to take both characters ST for the

signature,

otherwise if only the S, my program would regard the S of the unstable US,NT,+NNNNNN as prefix to valid data. The S could be left unstripped,

Re: It is possible for 89C51 to receive 7-bit RS232 data?

Re: It is possible for 89C51 to receive 7-bit RS232 data?

but

the T had parity bit "1".

Again, "ANL A,#7Fh" could not work for stripping the T. So I tried changing the line

You DO need to strip the bit off here.

You are only interested in the lower 7 bits.

'S' = 0x53

'T' = 0x54

```
CJNE A,#54h,WaitForChar ; T signature? No, wait again for character  
to:
```

```
CJNE A,#0D4h,WaitForChar
```

This should work –

```
Waitfor_ST:
```

```
call FetchCharacter
```

```
cjne A,#'S',Waitfor_ST
```

```
call FetchCharacter
```

```
cjne A,#'T',Waitfor_ST
```

and my program worked!!!

I had no time to think about why the "ANL A,#7Fh" could not work, so I simply change it to "ANL A,#7Fh" which worked OK because I had to

quickly

finish my display. More over I had to further enable my program to

suppress

leading 0 for the 5 digit display.

Thanks Jim for your suggestion and attention.

```
call FetchCharacter
```

**YOU NEED TO CLEAR THE CARRY HERE AS THE SUBB INSTRUCTION  
IS – SUBTRACT WITH BORROW.**

Re: It is possible for 89C51 to receive 7-bit RS232 data?

Re: It is possible for 89C51 to receive 7-bit RS232 data?

>>> clr c <<<

Yes Jim, you are right. I have carefully read again the 89C51 instruction set, and realize that my mistake was ignoring the Carry Flag in the SUBB instruction. The datasheet says that : Subb A,#54h where A is 0C9h, actually leave the value 074h in the Accumulator and Carry and AC flag cleared but OV set. 0C9h minus 54h is 75h, and the difference is due to the Carry Flag being set before the operation.

So, in my code :

```
FetchCharacter:
jnb RI,FetchCharacter
clr RI
mov A,SBUF
anl A,#7Fh
ret
```

```
call FetchCharacter
subb A,#30h
mov DPTR,#Table1
movc A,@A+DPTR
```

.  
.

I should insert "CLR C" before the line "SUBB A,#30h". I want to modify my program and test it on the display, but unfortunately yesterday I had sent the display to my customer. My last program with "ANL A,0Fh" worked because I didn't need to employ the SUBB instruction.

Thank you very much Jim, I guess I have to learn and practice a lot more.

Jusman

.