+ Reply to Thread

Results 1 to 8 of 8

Thread: nrf24L01+ and Picaxe 20x2 - how to read Registers

11-11-2010, 11:21 #1

hugoxp1

Member
Join Date: Dec 2009
Location: Porto, Portugal
Posts: 47

nrf24L01+ and Picaxe 20x2 - how to read Registers

Hi,

I want to interface the nrf24L01+ (http://www.nordicsemi.com/index.cfm?...display&pro=94) with a Picaxe 20x2 and I've started reading this tutorial made by Brennen http://www.diyembedded.com/tutorials...tutorial_0.pdf

This tutorial is C language oriented and I want to build my own functions (in programming Editor) for Picaxe 20x2

At the end of page 6 of this Tutorial Brennen wrote:

"First, you would bring CSN low and then send the command byte '00010000' to the 24L01. This instructs the 24L01 that you want to read register 0x10, which is the TX_ADDR register. Then you would send five dummy data bytes (it makes absolutely no difference what the data bytes contain), and the 24L01 will send back to you the contents of the TX_ADDR register" Finally, you would bring the CSN pin back high. All totaled, you will receive six bytes. When you send any command byte, the 24L01 always returns to you the STATUS register.

My problem is to understand when do I need to use "hspiin" and "hspiout" to obtain the response to a "instruction"...

e.g - To read “TX_ADDR”... is this statement correct? am I using hspiout and hspiin correctly?

Code:

```
READTXADDR: {

low cs                          ; enable chip select
hspiout (%00010000)             ; read register 0x10 = TX_ADDR
hspiout (1,2,3,4,5)             ; send five dummy data
hspiin(b1,b2,b3,b4,b5,b6)                   ; obtain the 5 bytes address
high cs                                ; disable chip select

}
```

If it's not asking to much... what will be the code to read the CONFIG register to "b0" variable?

(I really need to understand how to run these two commands to be able to forward next step...)

Thank you.

Last edited by hugoxp1; 11-11-2010 at 11:54.

Blog this Post | Reply | Reply With Quote

11-11-2010, 11:55 #2

hippy

Technical Support
Join Date: Jan 1970
Location: UK
Posts: 22,443

The code looks right in that it seems to match what is described; what happens when you try it, does it work as expected ?

Do you have access to a scope or a logic analyser ? If so that makes it much easier to see what's happening on the bus lines, otherwise it's a case of analysing or guessing and trying what needs to change.

I'm not sure how you mean by "what will be the code to read the CONFIG register to "b0" variable?"; this seems to read the CONFIG register, which returns six bytes.

Blog this Post | Reply | Reply With Quote

11-11-2010, 12:29 #3

hugoxp1

Member
Join Date: Dec 2009
Location: Porto, Portugal
Posts: 47

I've edit the post... the adress 0x10 is the TX_ADDR.

For CONFIG the adress is 0x00 (as you can see in the page 9).

I think I will have to write something like:

Code:

```
READ_CONFIG:{

        low cs          ; enable chip select
        hspiout (%00000000) ; send instruction register 0x00 = CONFIG
        hspiout (1)  ; send dummy byte
        hspiin(b0) ; obtain the value of the CONFIG register
        high cs

}
```

I've always to use 2 hspiout commands and 1 hspiin right?

note: I have bought 2 nrf24L01+ but i'm still waiting for the arrive and make all this experiments.

thank you

Blog this Post | Reply | Reply With Quote

11-11-2010, 12:34 #4

Chavaquiah

Senior Member
Join Date: Jan 2010
Location: Sintra, Portugal
Posts: 402

Well, we're both trying to come to grips with the nRF24L01...

I haven't yet tried to control it from a Picaxe so can't help you much with it. However, and from the code I've been writing for an AVR, I doubt it will work that way. SPI communcation is truly bidirectional. You're sending data at the same time it is responding.

I don't think you'll have much luck reading the status byte that is sent back while you're pushing out a command, unless you go bit banging.

As for reading a register, the nRF will answer back while you're sending the dummy bytes. You'd better be in hspiin

mode by then. Try instead:

```
hspiout (%00010000) ; read register 0x10 = TX_ADDR
hspiin(b2,b3,b4,b5,b6) ; obtain the 5 bytes address
```

Blog this Post | Reply | Reply With Quote

11-11-2010, 12:45 #5

hippy

Technical Support
Join Date: Jan 1970
Location: UK
Posts: 22,443

A further glance through the tutorial and the use of these dummy bytes make more sense.

What the tutorial is saying is to read the five ADDR bytes, you need to clock those bytes in, and to achieve that one can send five dummy bytes. This is because the device's SPI protocol sends bytes as it is receiving bytes.

To read ADDR it's probably ...

```
low cs
hspiout (%00010000)
hspiin(b1,b2,b3,b4,b5,b6)
high cs
```

To read CONFIG it's probably ...

```
low cs
hspiout (%00000000)
hspiin(b0)
high cs
```

I'm not sure you have the right SPI Mode; it may be SPIMODE10 or SPIMODE10e.

Blog this Post | Reply | Reply With Quote

11-11-2010, 12:54 #6

hippy

Technical Support
Join Date: Jan 1970
Location: UK
Posts: 22,443

> Originally Posted by **Chavaquiah**
> I don't think you'll have much luck reading the status byte that is sent back while you're pushing out a command, unless you go bit banging.

It may be possible to issue the HSPIOUT to send a byte command and then PEEKSFR SSPBUF to obtain what was passed back at the same time, which will be the status bits. You'd have to test that.

Blog this Post | Reply | Reply With Quote

12-11-2010, 22:31 #7

hugoxp1

Member
Join Date: Dec 2009
Location: Porto, Portugal
Posts: 47

> Originally Posted by **hippy**
> It may be possible to issue the HSPIOUT to send a byte command and then PEEKSFR SSPBUF to obtain what was passed back at the same time, which will be the status bits. You'd have to test that.

Hippy, can you explain me how to obtain this value: "SSPBUF"? I've tried to search all over and can't figure out how to do it...

thank you

Blog this Post | Reply | Reply With Quote

12-11-2010, 23:40 #8

hippy

Technical Support
Join Date: Jan 1970
Location: UK
Posts: 22,443

You need to look in the appropriate datasheet for the underlying PICmicro, then under Memory Organisation find the address of SSPBUF. For the X2 PICAXE's this is $FC9, or $C9 considering only the 8-lsb's of the address.

Then this should work ...

```
Symbol SSPBUF = $C9

HSpiOut ( $xx )
PeekSfr SSPBUF, b0
```

Here's a short demo program for the 20X2. Connect SPI SDO to SPI SDI on the X2, ignore SPI CLK. This sends a byte, reads the value in SSPBUF (it's what was simultaneously sent and recirculated back via the SDO to SDI wire), displays, it chooses next number and so on ...

Code:

```
#Picaxe 20X2
#No_Table
#No_data
#Terminal 9600

HSpiSetup SPIMODE10e, SPISLOW

Symbol SSPBUF = $C9

Do
```

Blog this Post | Reply | Reply With Quote

+ Reply to Thread

Quick Navigation Active PICAXE Forum Top

« Previous Thread | Next Thread »

Tags for this Thread

None
View Tag Cloud

Add / Edit Tags

Bookmarks

Digg
del.icio.us
StumbleUpon
Google

Posting Permissions

You may post new threads
You may post replies
You may post attachments
You may edit your posts

BB code is On
Smilies are On
[IMG] code is On
[VIDEO] code is On
HTML code is Off

Forum Rules

-- Picaxe Theme -- English (US)

Contact Us Main PICAXE Website Archive Top