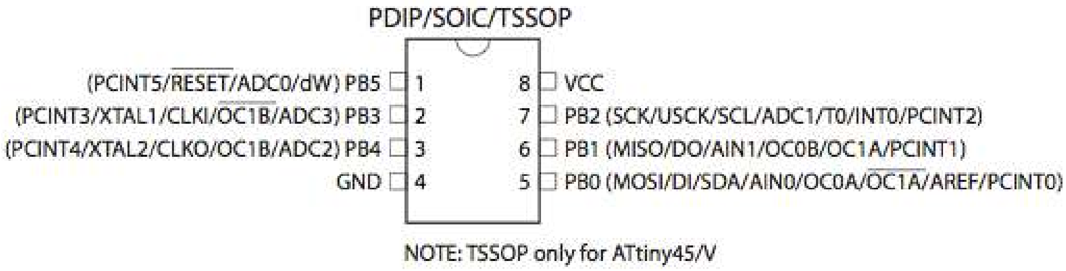
The Wandering Engineer

Pin Change Interrupts on ATtiny85

Pin Change interrupts on the Arduino have eluded me for awhile so when I finally figured them out last week, I also started messing with Pin Change Interrupts on the ATtiny. Unlike with the ATmega chips, examples are much harder to find for the ATtiny interrupts and even when you find them, I don't think the quality is quite as good. So here is a simple guide to working with Pin Change Interrupts on the ATtiny85.

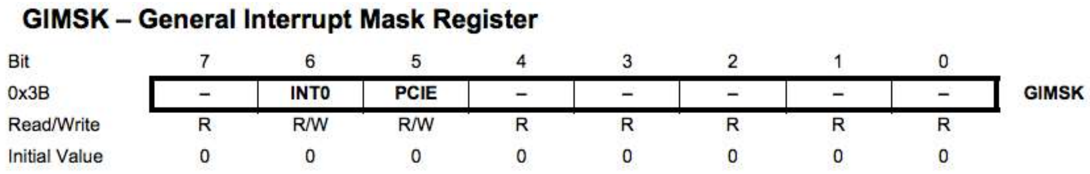
If you are reading this and aren't already comfortable with the idea of Pin Change Interrupts, please read this article **here (index.php?p=1135)** that describes how to use them on the ATMEGA328 and also goes into a lot of detail about what they are and how they're different than External Interrupts (and also why that name is a bit of a misnomer).

In case you don't want to read that article, I'll reiterate the basic rules of ISRs (Interrupt Service Routines): 1) Be as short as possible. 2) Don't use delays in the ISR. 3) Use volatile global variables if you're going to change them in the ISR.


PDIP/SOIC/TSSOP
(PCINT5/RESET/ADC0/dW) PB5 1
(PCINT3/XTAL1/CLKI/OC1B/ADC3) PB3 2
(PCINT4/XTAL2/CLKO/OC1B/ADC2) PB4 3
GND 4
8 VCC
7 PB2 (SCK/USCK/SCL/ADC1/T0/INT0/PCINT2)
6 PB1 (MISO/DO/AIN1/OC0B/OC1A/PCINT1)
5 PB0 (MOSI/DI/SDA/AIN0/OC0A/OC1A/AREF/PCINT0)
NOTE: TSSOP only for ATtiny45/V

(https://mwwalk.files.wordpress.com/2013/05/attiny-pinout.png)

Now, unlike the ATMEGA328 that the other article looked at, the ATtiny85 only has one port (Port B). And instead of using the command PCICR register to turn on interrupts, we'll use the GIMSK register. If you look at this register in the ATtiny85 datasheet located **here (http://www.atmel.com/images/atmel-2586-avr-8-bit-microcontroller-attiny25-attiny45-attiny85_datasheet.pdf)**, you'll see that there are two bits we can set. Bit 6 turns on External Interrupts while bit 5 turns on Pin Change Interrupts. Just put a 1 in each of the bits you want to turn on.

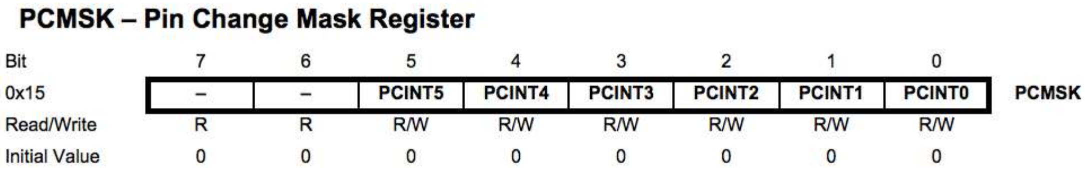


GIMSK – General Interrupt Mask Register

Bit	7	6	5	4	3	2	1	0	
0x3B	–	INT0	PCIE	–	–	–	–	–	GIMSK
Read/Write	R	R/W	R/W	R	R	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	

(https://mwwalk.files.wordpress.com/2014/08/atinty85-gimsk.png)

Next, we need to use a mask to say which pins we want triggering this Pin Change ISR. Just like with the ATMEGA328 example, we do this by setting the bits in the PCMSK register. Bits 0 through 5 controller the 6 pins that can be used for outputs. A 1 in that bit means an interrupt can occur on that pin, while a 0 means it cannot. Remember that whenever any of the pins are triggered it will call the same ISR. Finally, you need to write the ISR function. Because there is only one port on this chip, there is only one function you need to worry about and it is called ISR(PCINT0_vect){}.



PCMSK – Pin Change Mask Register

Bit	7	6	5	4	3	2	1	0	
0x15	–	–	PCINT5	PCINT4	PCINT3	PCINT2	PCINT1	PCINT0	PCMSK
Read/Write	R	R	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

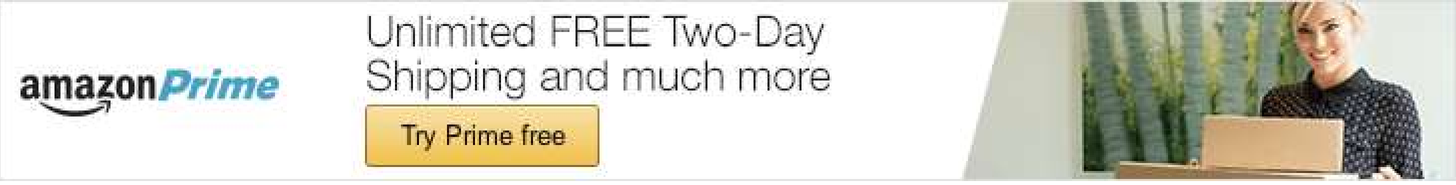
(https://mwwalk.files.wordpress.com/2014/08/attiny85-pcmsk.png)

Putting it all together

Here is a sketch that has all of these put into place.

```
#include &quot;avr/interrupt.h&quot;

volatile int value=0;

void setup()
{
    GIMSK = 0b00100000;    // turns on pin change interrupts
    PCMSK = 0b00010011;    // turn on interrupts on pins PB0, PB1, &amp;amp; P
    sei();                 // enables interrupts
}

void loop()
{
}

ISR(PCINT0_vect)
{
    value = 1;             // Increment volatile variable
}
```

a



Posted on **August 11, 2014June 27, 2016** by **mwwalk**Posted in **Electrical**, **Engineering**, **Software**, **Tutorials**Tagged **Arduino**, **Attiny**, **ATtiny85**, **AVR**, **Interrupt**, **Pin Change Interrupt**.

Related

Arduino Pin Change Interrupts In "Electrical"

Rotary Encoder on the ATtiny85 - Part 2 In "Electrical"

ATtiny i2c Slave In "Engineering"

22 thoughts on “Pin Change Interrupts on ATtiny85”

1. **MIKE SAYS:**
   August 21, 2014 at 5:19 am
   It’s GIMSK, not GISMK!

   □ **Reply**

   MWWALK SAYS:
   August 21, 2014 at 3:20 pm
   Good Catch! It’s fixed now. Thanks

   □ **Reply**

   1. **UNIXGURU2013 SAYS:**
      January 19, 2015 at 7:27 am
      Any chance I can get some contact info to properly credit you in an upcoming blog?
      I’m using most of your Quadrature Decoder solution in a closed loop DC motor controller, and would love to credit the origin.
      Appreciated.

      MWWALK SAYS:
      January 20, 2015 at 2:55 pm
      I’m Matt Walker and a grad student in the Intelligent Robotics Lab at the University of Oklahoma in Norman, OK. You can credit that or this blog, either is fine with me. As I’ve mentioned before everything on this site is under the beer license. That means that it’s free to use as long as credit is given and if you find it really useful and we ever meet, you should buy me a beer. 🙂

   2. **SONOSAMP SAYS:**
      June 27, 2016 at 12:22 pm
      It’s not fixed now. It says “MIGSK” now…

      MWWALK SAYS:
      June 27, 2016 at 4:10 pm
      Thanks SONOSAMP. I had corrected it in the code but not the text.

2. **JEAN-ROCH BLAIS SAYS:**
   September 19, 2014 at 5:04 pm
   I’m fairly new to arduine, but I like and understand the way you code in binary:

   ```
   void setup()
   {
   GIMSK = 0b00100000; // turns on pin change interrupts
   PCMSK = 0b00010011; // turn on interrupts on pins PB0, PB1, & PB4
   sei();
   ```

   so here is my lamo question …

   where are GIMSK, PCMSK defined. I couldn’t find in which library … would you know ?
   It would help to know the list of predefined system register variables. I can see that the hardware documentation gives the same names, I could blindly use that… but I’d much prefer to see the software definition !

   Your interrupts instructables are very good, and allowed me to make a huge step in understanding attiny85 interrupts. By the way I’m only using the atmel ISP mkII, it is still available here in Canada for a mere 40$ from **http://www.futureelectronics.com..**.

   Thank’s
   jrb.

   □ **Reply**
3. **JEAN-ROCH BLAIS SAYS:**
   September 19, 2014 at 5:15 pm
   Sorry bad link for future electronics, use **http://www.futureelectronics.com/en/Search.aspx?dsNav=Ntk:PartNumberSearch**|atavrisp2|1|,Ny:True,Nea:True instead…

   □ **Reply**
4. **JEAN-ROCH BLAIS SAYS:**
   September 21, 2014 at 9:37 pm
   Ok I found GIMSK, PCMSK… software definitions:

   In avr-libc/user-manual/io_8h_source.html (io.h) we find:

   “379 #elif defined (__AVR_ATtiny85__)”
   “380 # include ”

   going to “avr/iotn85.h” , there we find an “# includes avr/iotnx5.h”

   and again, going to “avr/iotnx5.h”, we finally find the definitions of GIMSK, PCMSK, etc…

   You probably knew all of that, still, I’m writing this in case someone like me might be wondering too 🙂 !!!

   Thank’s for this forum !

   □ **Reply**
5. **JOHN SAYS:**

January 7, 2015 at 2:05 am
Thank you for your excellent explanation. Suppose I want to use interrupts for my RF433 receiver (data pin of de RF433 receiver connected to pin#2 of the ATTiny85), is that the way to go? When I do that (at328p), the interrupt gets triggered all the time due to the RF noise. This seems very inefficient. However I'd like to have my RC commands priority over everything else. This may be offtopic but I reckon you might have an educated opinion on this. Thanks again!

□ **Reply**

MWWALK SAYS:
January 20, 2015 at 2:57 pm
My first thought would be to filter the signal of the data pin. Look up an RC filter which is pretty simple and cheap. Alternatively, you could use a resister as a pullup or pulldown on the line. Good luck!

□ **Reply**

1. **YC SAYS:**
July 4, 2017 at 8:08 pm
John, I am trying to do exactly the same and encountered the same problem of RF433 receiver data pin triggering the interrupt all the time. Have you figured out a good solution? Thanks for your help.

□ **Reply**

6. Pingback: **Freelance: Pixel Mosaic | Thomas Tesla Freelancing**
7. **DAVE SAYS:**
March 5, 2015 at 12:48 am
Most excellent! This allowed me to fix a last minute "one line change" I made before ordering some boards. I swapped PIN2 and PIN0 to make board layout easier not even thinking about the interrupt issue and the board didn't work because what was happening on PIN2 was triggering INT0 (Dooh!). This provided me with the knowledge to fix the hardware issue in software using PCINT0 and pin masks. Thanks for taking the time to write this.

□ **Reply**

8. **DARLINGTON SAYS:**
August 12, 2015 at 11:22 am
Nice one, How do I detect which of the pin triggered the interrupt? and how do you set the triggering edge (going low or going high)

□ **Reply**

MWWALK SAYS:
August 19, 2015 at 9:14 pm
This method isn't really meant to figure out which pin triggered the interrupt since all the pins on the port could have triggered it. To figure out which pin triggered the interrupt, you'd have to check the status of each pin and compare it to a previous status. Keep it mind though that it should only trigger for changes on pins that you set with PCMSK. You can't really set whether it triggers when RISING or FALLING; since it is a pin change interrupt, it will trigger on any change.

□ **Reply**

9. **DARL SAYS:**

August 20, 2015 at 8:15 pm
Well understood, Thanks

□ **Reply**

10. Pingback: **ATtiny85 « Silas blog**
11. **ELDON TENORIO SAYS:**
July 22, 2017 at 5:43 pm
Been looking for something like this especially for ATtiny85! Great job there. Will it be ok to mention this page when I reference this in my article regarding interrupts with Attiny85?

Thanks!

□ **Reply**

MWWALK SAYS:
August 9, 2017 at 5:38 pm
Sure, reference away!

□ **Reply**

12. Pingback: **Freelance: PixelMosaic - MikeThomasMakes**
13. **CHARLYCOP SAYS:**
May 24, 2018 at 8:29 am
Hi everybody,

I'm new in the ATtiny85 and even microcontroller area.
I program my ATtiny85 via my Arduino UNO and the Arduino IDE.

What don't understand, if I should write my program in the real ATtiny syntax or can I use the simple arduino "language" ?

I'm asking this because I have some trouble to configure an external interrupt. A simple interupt by a switch in order to change the value of a global bool value.

Can you help me ?

□ **Reply**