

The Toivonen's Blog

Wednesday, December 9, 2015

Fixing Pin P5 or 6 on Digispark Clones

The digispark ATTINY85 is a handy microcontroller for simple projects. The genuine version, available here: <http://digistump.com/products/1> comes with the micronucleus bootloader installed and 6 GPIO pins enabled.

There are Chinese clones available on ebay for about \$1.50 that are pin for pin compatible and come with the micronucleus bootloader installed. The catch is that pin P5 (the last one, physically counted as number 6, ADC0) may be configured as a reset pin. If you only need 5 pins, or want a reset pin that's great, you can stop reading. If you need 6 pins, read on.

The ATTINY85 comes from the factory set up for ISP (in system programming), for this to work it needs a reset pin. The micronucleus bootloader allows the user to program the ATTINY85 with a USB connection. This is well documented on the digistump site.

After the bootloader is added, we no longer need that reset pin, and on the genuine boards they go ahead for us and set the "fuse" on the ATTINY85 to disable the reset pin, allowing it to be used as a GPIO pin. On the clones they may not.

To test if your clone can use pin 5, there are some simple methods. One would be to upload a sketch that blinks an led on pin 5, if it works that's great! You don't need to change anything. If it doesn't work pin 5 will not work for GPIO.

Another simple test would be to pull pin 5 to ground through a small resistor (~220 ohms) while a program is running. If the program resets (the micronucleus bootloader waits 5 seconds and then the program runs) you cannot use pin 5 for GPIO.

(If pin 5 is working for GPIO and you want change it to a reset pin, it is possible. You need a high voltage programmer, which is fairly simple to build and use, but it beyond the scope of this post. Look here to get started: <http://www.rickety.us/2010/03/arduino-avr-high-voltage-serial-programmer/>)

Ok, so now you know that your pin 5 is configured as a reset pin and you want it for GPIO.

First and foremost be sure the micronucleus bootloader is installed on your ATTINY85 and working properly.

After we change the configuration fuse, you will not be able to load micronucleus without a high voltage programmer! Be sure the bootloader is present and working first. If you can upload programs to your ATTINY85USB via the USB connection from the Arduino IDE the bootloader is present and working.

You *will* still be able to upload programs via the USB connection as you normally do after changing the configuration fuse.

The next thing you will need is another Arduino, I used a Duemilanove, but you can also use an Uno. Using the Arduino IDE (I used v1.6.6), go to file > examples > ArduinoISP and upload this sketch to your Duemilanove (or Uno). If you have been playing around with your digispark, be sure the bootloader and board in the Arduino IDE are set correctly to the board you are now programming.

After uploading, go to tools > port and note the port where your Arduino is connected. For me it was '/dev/ttyUSB0'. We will need this later.

About Me

[TheToivonen](#)

[View my complete profile](#)

Blog Archive

► [2023](#) (3)

► [2021](#) (1)

► [2020](#) (2)

► [2019](#) (1)

► [2016](#) (1)

▼ [2015](#) (8)

▼ [December](#) (2)

[Fixing Pin P5 or 6 on Digispark Clones](#)

[New Mexico Balloon Festival](#)

► [October](#) (1)

► [September](#) (1)

► [July](#) (2)

► [June](#) (1)

► [March](#) (1)

► [2014](#) (7)

If you are using a Duemilanove, add a 120 ohm resistor from RESET to 5v on the Duemilanove board. Do not use a resistor smaller than 110 ohms or larger than 124 ohms. *Please do not skip this step. Read the comments for more information.*

On an Uno, do not add the resistor but instead add a 10uF capacitor from RESET to GROUND on the Uno board. *Please do not skip this step. Read the comments for more information.*

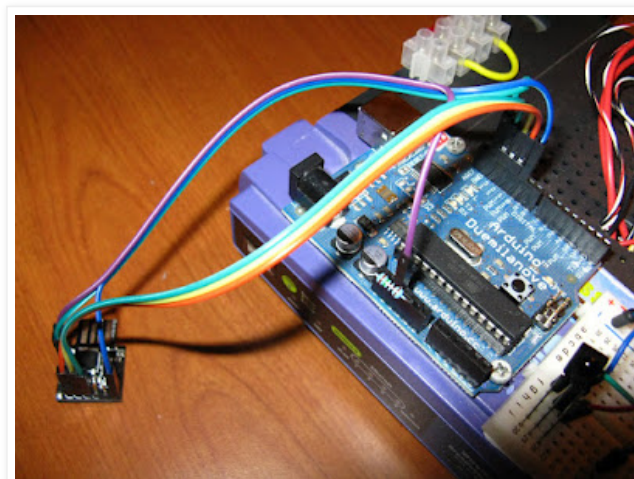
Connect as shown:

Duemilanove --> Digispark ATTINY85USB Clone

GND --> GND
 5v --> 5v
 10 --> P5
 11 --> P0
 12 --> P1
 13 --> P2

These are the connections for Duemilanove and ATTINY85, for other boards be sure to determine the proper pins, you want SS, MOSI, MISO, and SCK on the other Arduino connecting to RESET (P5), MOSI (P0), MISO (P1), and SCK (P2) on the digispark ATTINY85USB clone. Note these are not the pins on the ATTINY chip itself, but the IO pins on the digispark clone.

It should look something like this...



Next go to a terminal and enter (replace '/dev/ttyUSB0' with the port your Arduino connects to that you noted earlier):

```
avrdude -P /dev/ttyUSB0 -b 19200 -c avrisp -p attiny85 -n
```

It should output something like this...

```
avrdude: please define PAGEL and BS2 signals in the configuration file for part Attiny85
avrdude: AVR device initialized and ready to accept instructions
```

```
Reading | ##### | 100% 0.02s
```

```
avrdude: Device signature = 0x1e930b
```

```
avrdude: safemode: Fuses OK
```

```
avrdude done. Thank you.
```

I am using linux, if you use windows I would assume this works the same in command prompt, if you enter (replace '/dev/ttyUSB0' with the port your Arduino connects to that you noted earlier):

```
avrdude.exe -P /dev/ttyUSB0 -b 19200 -c avrisp -p attiny85 -n
```

If all goes well, the next command to enter is (replace '/dev/ttyUSB0' with the port your Arduino connects to that you noted earlier):

```
avrdude -P /dev/ttyUSB0 -b 19200 -p attiny85 -c avrisp -U hfuse:w:0x5F:m
```

The response should be:

```
avrdude: please define PAGEL and BS2 signals in the configuration file for part ATtiny85
avrdude: AVR device initialized and ready to accept instructions
```

```
Reading | ##### | 100% 0.02s
```

```
avrdude: Device signature = 0x1e930b
avrdude: reading input file "0x5F"
avrdude: writing hfuse (1 bytes):
```

```
Writing | ##### | 100% 0.02s
```

```
avrdude: 1 bytes of hfuse written
avrdude: verifying hfuse memory against 0x5F:
avrdude: load data hfuse data from input file 0x5F:
avrdude: input file 0x5F contains 1 bytes
avrdude: reading on-chip hfuse data:
```

```
Reading | ##### | 100% 0.01s
```

```
avrdude: verifying ...
avrdude: 1 bytes of hfuse verified
```

```
avrdude: safemode: Fuses OK
```

```
avrdude done. Thank you.
```

That's all, your digispark clone should now allow you to use P5 as a GPIO pin, and be programmable via the USB connection. If you want to go back to having a reset pin, or program with ICSP, you need a high voltage programmer. See the link above.

Posted by [TheToivonen](#) at 16:04

33 comments:

 Anonymous [22/2/16 14:45](#)

Thank you so much!

I spend the last weekend with a digispark, trying to get adc0 working without success. I want to use it as a adc over i2c.

Now it works as expected.

Thank you, thank you, thank you!

[Reply](#)



[TheToivonen](#) [22/2/16 15:04](#)

Glad to be of help!

[Reply](#)