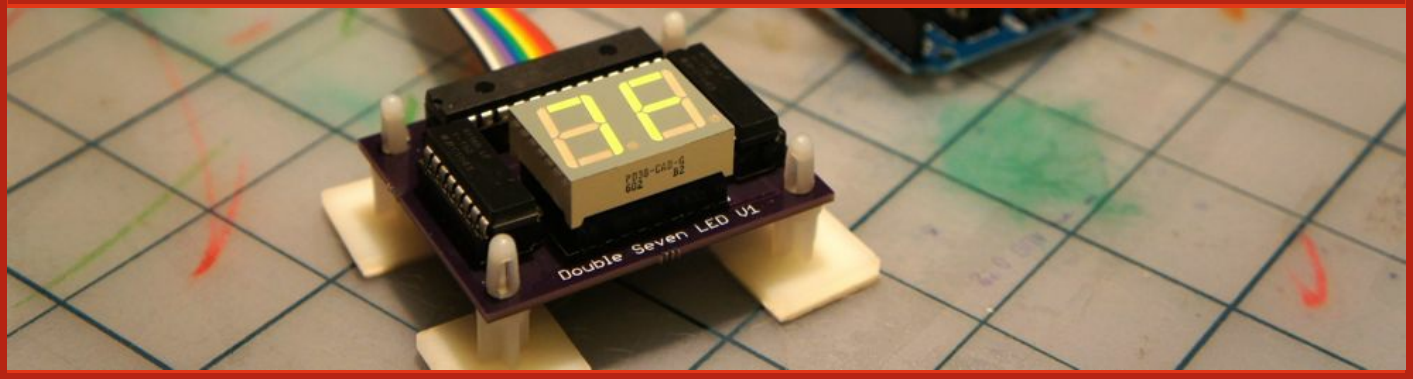


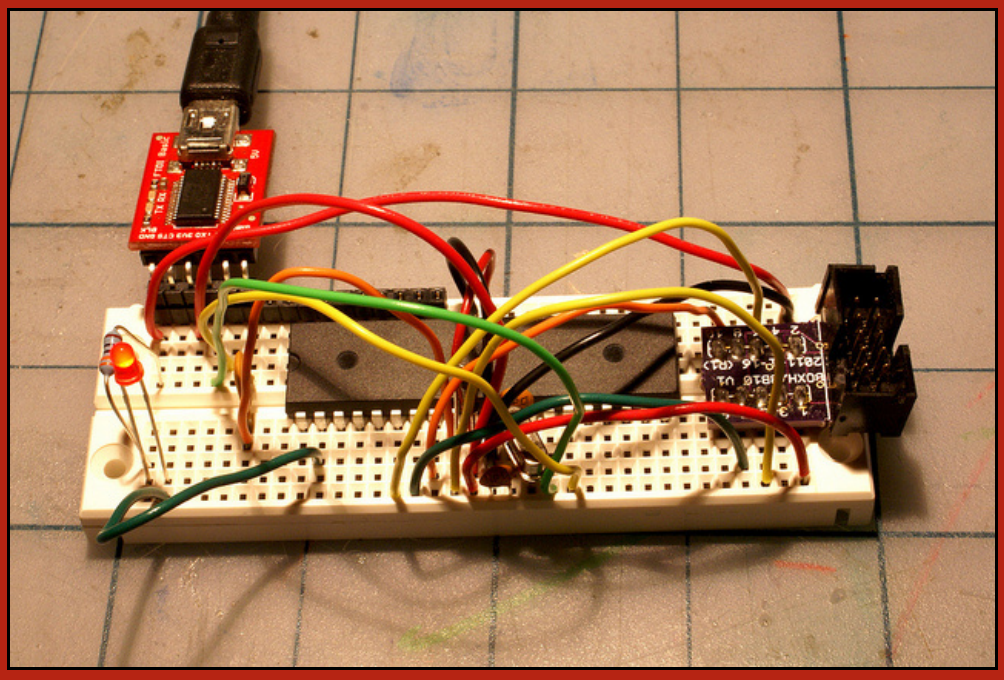
← Getting Started with nRF24L01+ on Arduino

nRF24L01+ Running on Maple →

NOVEMBER 27, 2011 · 12:45 PM

↓ Jump to Comments

Arduino on ATmega1284P



Sooner or later, the Arduino starts to feel a little claustrophobic. Your sketches start running out of memory, so you need more RAM. You want to talk serial to another peripheral (like an RFID Module) AND watch the action in the Serial Monitor at the same time, so you need more UARTS. You want to use SPI and control a motor and read a few sensors, and pretty soon you're out of pins, so you need more I/O pins. You're logging data, and running out of 1k EEPROM fast, so a bit more EEPROM would sure be handy.

The obvious solution to this problem is an Arduino Mega 2560. Oh yeah, this thing is powerful! 54 pins! 4 UARTS! 8k RAM! You add it to your cart, and then realize it's almost $50.

Now maybe you're wondering, isn't there anything in-between that might be a bit less cash? Indeed there is, it's call the ATmega1284P and today I'm going to explore getting Arduino to run on it, on a breadboard.

Comparison

Here's a quick breakdown of three chips... The 328P powers the Uno, the 2560P powers the Mega 2560, and the 1284P is the 'Goldilocks' chip,

nestled right in the middle–EXCEPT for RAM. It has an abundance of RAM (16k!) which is important because RAM is typically the resource you run out of the quickest.

Feature	328P	1284P	2560P
Price	$2.99	$4.66	$11.28
RAM	2k	16k	8k
Flash	32k	128k	256k
EEPROM	1k	4k	4k
UART	1	2	4
IO Pins	23	32	86
Interrupts	2	3	8
Analog Inputs	6	8	16

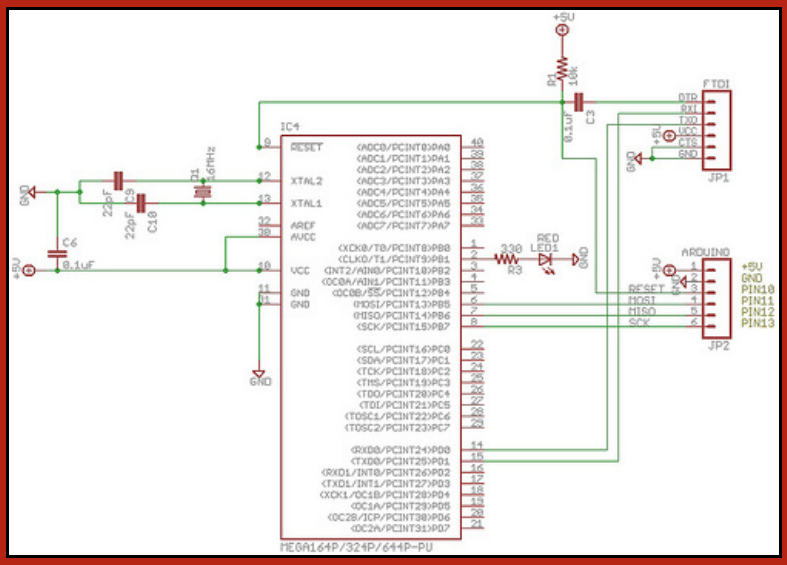
Background Reading

I'm going to build an Arduino ATmega1284P circuit on a breadboard, burn a bootloader, and upload sketches to it, all using the Arduino 1.0 IDE. This will all go better if you have done it once first with a regular ATmega328P. Doing these things on that chip is well-documented, and easy to get support on the forums if something goes wrong.

So first, read up and practice up on a 328P with help from these handy pages on the Arduino Wiki:

- Building Arduino on a Breadboard
- Arduino ISP Tutorial

Building the Circuit



There are three sections to the circuit show above. (Click on the image for a higher-resolution).

- Power/Timing. I'm using a 16MHz crystal and 22pF caps for timing and a 0.1uF cap for power.
- FTDI. I use a Sparkfun FTDI breakout to program everything, so I've set this breadboard up with a connector for that mechanism.
- Arduino ISP. To burn the bootloader using Arduino as ISP, six connections are needed from the Arduino. The from/to pins are indicated on the schematic.

For another view, check out this forum thread: Using the 1284p/664p (IDE, bread board and boot loaders)

Get the Software

The Arduino makes it easy to support a new 'Hardware Platform'. The files you need are hosted on github, the Mighty 1284P Platform for Arduino.

WARNING this ONLY WORKS on Arduino 1.0. It would be possible to back-port to 0022, but seems unneeded now that 1.0 is nearing final release.

To install it...

1. Download the ZIP File
2. Unzip it a folder called 'hardware' off your sketches directory, e.g. /Users/maniacbug/Source/Arduino/hardware/mighty-1284p
3. Restart the IDE
4. Select Tools > Boards > Mighty 1284P

Burn a Bootloader

Once you have the Mighty 1284P platform set up, burning a bootloader is trivial. Follow the Arduino ISP instructions. You first burn the "Arduino ISP" sketch onto a regular Ardunio. Then change the "Boards" setting to the Mighty 1284P, hook up the 'Arduino' connections in the schematic above, choose "Tools > Programmer > Arduino as ISP" from the IDE, and then "Tools > Burn bootloader."

To verify that it worked, move the LED to pin 1, and power cycle the breadboard. When the bootloader starts it flashes pin 1.

Upload a Sketch

Uploading a sketch is trivial now that the platform is set up and bootloader is in place. Try the 'Blink' example first, but change the LED pin to '1' in the sketch. (Of course, move the LED back to the right pin, which is pin 2 on the MCU).

Understanding Pin Mappings

When you refer to a "pin" in the Arduino environment, e.g. "digitalWrite(13,HIGH)", the system maps that pin onto a MCU pin. There is definitely no standard pin mapping for 1284P, many people have published their own. I chose a mapping that makes sense for the breadboard, where the digital pins start at MCU pin 1, and simply go around the chip counter-clockwise.

```
                    +---\/---+
          (D 0) PB0 1|        |40 PA0 (AI 0 / D24)
          (D 1) PB1 2|        |39 PA1 (AI 1 / D25)
      INT2 (D 2) PB2 3|        |38 PA2 (AI 2 / D26)
       PWM (D 3) PB3 4|        |37 PA3 (AI 3 / D27)
    PWM/SS (D 4) PB4 5|        |36 PA4 (AI 4 / D28)
      MOSI (D 5) PB5 6|        |35 PA5 (AI 5 / D29)
  PWM/MISO (D 6) PB6 7|        |34 PA6 (AI 6 / D30)
   PWM/SCK (D 7) PB7 8|        |33 PA7 (AI 7 / D31)
                 RST 9|        |32 AREF
                VCC 10|        |31 GND
                GND 11|        |30 AVCC
              XTAL2 12|        |29 PC7 (D 23)
              XTAL1 13|        |28 PC6 (D 22)
      RX0 (D 8) PD0 14|        |27 PC5 (D 21) TDI
      TX0 (D 9) PD1 15|        |26 PC4 (D 20) TDO
RX1/INT0 (D 10) PD2 16|        |25 PC3 (D 19) TMS
TX1/INT1 (D 11) PD3 17|        |24 PC2 (D 18) TCK
     PWM (D 12) PD4 18|        |23 PC1 (D 17) SDA
     PWM (D 13) PD5 19|        |22 PC0 (D 16) SCL
     PWM (D 14) PD6 20|        |21 PD7 (D 15) PWM
                    +--------+
```

Alternatives

There are a handful of folks who have built a fully-functional board with 1284P. These are all worth checking out to see if one meets your needs. Still, I think there is a gap to be filled for a cheap 1284P-based Arduino with minimal add-ons.

- Sanguino. The classic. Uses ATmega644 which is real close to the 1284, but the core is not maintained beyond Arduino 0017.
- Bobuino. Capable, fully-featured board, with SD, RTC, USB, and more. You pay $80 for all those features–which is great if you need it all, but it's too expensive if you just need a bit more power for your projects. In my view, you might as well get a Mega 2560. My other complaint is the lack of coherent software distribution. The (hidden) directions expect users to cobble together a system from various places, and then includes a (hidden) package of files which is not under source control. And worse, no source for the bootloader.
- Brewtroller Sanguino1284P.zip. Looks like the most complete distribution. Not ported to 1.0 yet. Uses the backwards analog pin mappings from avr-developers. Designed for specialized use (brewing control).
- Calunium Photos on Github. Ported to 1.0. Uses a rather custom pin arrangement. No bootloader. Still, looks like the best bet overall.

- BahBots Controller. See Setting up support for the BahBots controller in Arduino. I couldn't find the bootloader, it runs at 20MHz (I want 16MHz). Again, requires you to cobble together a core using the avr-developers core.

Related

125Khz RFID Module RDM630
In "Arduino"

Getting Started with nRF24L01+ on Arduino
In "Arduino"

Arduino on Ice: Internet Radio via Shoutcast
In "Arduino"

Filed under Arduino 158 Comments

158 responses to "*Arduino on ATmega1284P*"

Mark
December 1, 2011 at 3:01 pm

Hi,

Nice work on this, any chance you could add a 1284P 8Mhz boot loader version as I am interested in running this chip at 3.3v?

Reply

maniacbug
December 1, 2011 at 8:27 pm

Thanks. Ok, I put up a 8MHz board. Please test it out. BTW, you can almost certainly run at 16MHz even though it's technically 'overclocking' it. I have 328p's running 3.3V at 16MHz for months at a time.

Reply

Mark
December 2, 2011 at 12:02 am

Thanks,

Will give it s go this weekend along with another I am trying.

I notice boards.txt has 2 16mhz boards is it just a case of modifying the relevant board name?

maniacbug
December 2, 2011 at 6:25 am

Oh yeah, copy/paste error.[illegible] Forgot to change the title.

Mark
December 2, 2011 at 10:11 am

Ok, downloaded and first thing I notice is that the 8Mhz board is the only one that is listed in the IDE, must be something else wrong with the boards.txt file for the 16Mhz one.

Anyway I have successfully burned the 8Mhz boot loader, but when I try to upload sketches via ftdi on TX0/RX0 I just get not in sync 0x94, does the upload baud rate etc need to be reduced in the boot loader for the 8Mhz version?

Thanks,
Mark

Reply

maniacbug
December 2, 2011 at 9:34 pm

Ok I did change the baud rate, and finally tested it myself on 8MHz. Works great now for me.

Reply

whirleyes
December 3, 2011 at 6:25 am

thanks! With few mods, the classic Sanguino 644p works too.

Reply

maniacbug
December 3, 2011 at 8:35 am

Sweet. Do you want to submit the mods so others can get it too?

Reply

Another Mark
December 20, 2011 at 7:23 am

Very nice work on this maniacbug. I am also asking about these mods for 644P and bootloader for it.

whirleyes
December 20, 2011 at 10:46 pm

Here's my forking of yours.. https://github.com/whirleyes/mighty-1284p.
I did a compare of yours with arduino v1.0 release. Based on the changes, I've made some changes to add support for Atmega 644p.

Len
May 20, 2013 at 8:24 am

Are you actually ported MightyBoot to 644p i.e. this is different bootloader and pin definitions from Sanguino? I'm asking because Sanguino has incorrect pin defs (i.e. some Analog ports are simply not working, problems with internal pullups and interrupt pins). I've been looking for an alternative bootloader that will run on 644p for a while now

Reply

TheLoon
December 9, 2011 at 9:40 pm

So which 1284p should I be looking for? If I use the 10mhz version do you think it would be possible to run it at 16mhz powering it at 3.3 v therefore elimitnating the need for a separate 3.3v LDO voltage regulator an LLC with 3.3v sensors? Just wondering????

Reply

maniacbug
December 9, 2011 at 10:21 pm

I say get the 20MHz version, run it at 16MHz, and give it 3.3V. Doing it without voltage regulator will be fine. It's slightly 'overclocking' at that speed. You could also go down to 8MHz if you wanted to be safe.

Reply

TheLoon
December 10, 2011 at 11:19 am

Thanks, I'll order the ATMega1284p-pu 20Mhz but can you verify that the chip does indeed have 8 PWM pins as per your pin map? Any chance you have a tested eagle lbr for the chip and it's prospective packages? Thanks again.

Reply

maniacbug
December 12, 2011 at 6:33 am

This blog makes no warranties ☺ ☺ ☺

Reply

maniacbug
January 16, 2012 at 9:48 am

"/usr/share/arduino-1.0/hardware/arduino/variants/standard/pins_arduino.h" tells me you're not using the files from https://github.com/maniacbug/mighty-1284p .

Also. it looks like you're using a custom build process? Have you gotten it to work in the IDE first?

Reply

Timothy Reaves
January 16, 2012 at 9:59 am

Really? https://github.com/maniacbug/mighty-1284p/blob/master/variants/standard/pins_arduino.h is the file in question. So is that file there, and not supposed to be used anywhere? I can try to pull it down again.

Yes, I'm using arduino-cmake, and no, I've no desire to use the very pathitic Arduino IDE. Whereas I very well may be doing something wrong with your code, arduino-cmake works well.

Thanks.

maniacbug
January 16, 2012 at 1:52 pm

> I've no desire to use the very pathitic Arduino IDE

That's understandable, it just makes it difficult for anyone else to help if we're not also using your build system. The best thing to do is get it working in the IDE, and then move it out to your own system.

> So is that file there, and not supposed to be used anywhere?

That file should be in "hardware/mighty-1284p/variants/standard". The file you're picking up is in "hardware/arduino/cores/arduino", which I'd expect to be the stock pins_arduino.h that comes with Arduino.

Timothy Reaves
January 17, 2012 at 6:14 am

O.K., I copied the files over incorrectly. I copied over the content of the directory, not the mighty-1284p directory itself. And for that, unfortunatly, arduino-cmake will not work, as it expects only /hardware/arduino, not hardware/mighty-1284p.

Thanks for helping me figure the layout out.

IgorChemij
January 22, 2012 at 8:52 am

Hi,

I would like to run my 1284p with 20MHz on 5V. I need to boot it over UART-FTDI (like Arduino) but I do not plan to use Arduino environment. Would you please, suggest me if I change Makefile of bootloader (16->20MHz two entries) it should be enough to do this?

Thank you!

Reply

maniacbug
January 22, 2012 at 9:15 am

It seems reasonable that it would work that way. You might need to reduce the baud rate, though. In any case, you might as well try it out and see what happens. However, since so few people run at 20MHz, you might want to consider 16MHz just because you can leverage more of other peoples' efforts.

Reply

retrolefty
January 25, 2012 at 6:16 am

Nice blog. I think the mega 1284 is a great chip and most likely the 'last stand' for us DIP package fans. My only suggestion is you consider changing the word RAM to SRAM in this blog, as the three formal memory names used in these AVR chips are FLASH, SRAM, and EEPROM, and I would suggest using the names per datasheet usage would be 'more better'. 😉

Reply

maniacbug
January 25, 2012 at 9:19 am

Thanks! And thanks for the suggestion. That makes sense.

Reply

Flo
January 29, 2012 at 8:54 am

Now that is exactly what i was looking for!
24 Digital pins really sound like Christmas 🙂
I'll just hope that i get the bootloading process right…

I was about to order a Mega for my project but now it looks like i will get away with it paying only a fraction for having some extra pins!
Thanks for sharing!

Reply

maniacbug
January 29, 2012 at 9:08 am

Cool, glad to hear it! Be sure to get the latest platform from github as of a minute ago. I just pushed up a new bootloader using optiboot which is MUCH better. And I'd love to get some more testing on it.

Reply

Flo
January 30, 2012 at 1:58 am

Optiboot!
Love to hear that 🙂
Wont have time to test it until next week but I'll keep you updated.

Alkex Smith
February 7, 2012 at 8:19 am

Hi ManiacBug,
A big thanks is in order! I've been experiencing that claustrophobic sensation for months, hiving to drop an idea out of my sketch for every new idea that pops up. Can't wait to have 4x the flash and spare no expense 🙂

Couple quick questions: I'm using the following libraries with my Atmega328 (arduino ide 1.0):

#include (external 1025k eeprom)
#include (external 1025k eeprom)
#include (external 1025k eeprom)
#include (configured a pin to output 1MHZ)
#include (time critical stuff)

Do you have any reason to believe that these won't work with the atmega1284p?
Would be a deal breaker for me…

I've ordered a couple already but thought I'd run it by you.

THANKS!!!
Alex

Reply

Alkex Smith
February 7, 2012 at 8:21 am

something went weird with the post, the includes are:
Arduino.h
Wire.h
E24C1024.h
TimerOne.h
MsTimer2.h

Thanks

Reply

maniacbug
February 7, 2012 at 9:17 am

If it works with 328p, it will work with 1284p. The timers are interesting, though, they may require some changes to work with 1284p. You MAY be the first person to use them on this chip. If you run into trouble, post to the Arduino forums. There are a handful of people using 1284p now, and one of them may have used the timer libraries.

Reply

Steve Marple
February 7, 2012 at 9:56 am

I'm using the timer2 with my Calunium board, which also uses the '1284P. There are a few example sketches in the Calunium Github repository: https://github.com/stevemarple/Calunium/tree/master/software/examples. They are written to work with the on-board real-time clock but there should be enough comments for you to make it work with another board and clock source. I haven't tried MsTimer2.

Alkex Smith
February 7, 2012 at 11:02 am

Thanks MB, Steve, I look forward to experimenting 🙂

Reply

Stuart
February 10, 2012 at 7:45 am

Not having much luck getting the bootloader on using arduino1.0 a deicimila with an atmega168 as my arduinoISP

avrdude: Yikes! Invalid device signature.
Double check connections and try again, or use -F to override
this check.

Any suggestions ?

Reply

maniacbug
February 10, 2012 at 9:19 am

Are you able to program other Arduino chips using this setup?

Turn on verbose output and paste the whole thing into pastebin and post a link here.

Is this a virgin 1284p that has never had anything on it? If so you likely have something booked up wrong.

Reply

Stuart
February 12, 2012 at 3:05 am

Ok.

So, burning the arduinoISP onto a mega168 on a deicimila works ok.

Connect up to a deicimila with a 368 as per this diagram from the arduinoISP page (http://arduino.cc/en/uploads/Tutorial/arduinoisp.png) , and try to burn the bootloader fails here is the verbose logging on pastebin

http://pastebin.com/F8uBVM1J

So possibly a problem with the programmer ☹

maniacbug
February 12, 2012 at 8:02 am

This is a good point to turn to the Arduino forums for help. Lots of knowledgeable people there about these topics.

Stuart
February 13, 2012 at 8:12 am

I've posted details on the arduino forum http://arduino.cc/forum/index.php/topic,91767.0.html

Phil
February 11, 2012 at 5:07 pm

Hi maniacbug,

Your Mighty 1284p Platform seems to be very nice.

Maybe I'm posting a stupid question, but:
in your section "Burn a bootloader", when you say: " You first burn the "Arduino ISP" sketch onto a regular Ardunio. ", that "regular" Arduino can be an UNO or must be older version ?

Thanks.

Reply

maniacbug
February 11, 2012 at 6:26 pm

Thanks! UNO should work. There are plenty comments on the forums about what's needed to make UNO work. Me, I just use my own clones.

Reply

Phil
February 12, 2012 at 7:08 am

...using clones: okay, good idea! I can make a simple Arduino clone in breadboard only to burn bootloaders... Thanks for the suggestion.

Constantin
February 12, 2012 at 8:10 am

FWIW, Maniacbug, the Bahbots controller runs at 18.4MHz. As I recall, this rather rarely-used frequency had to do with timing issues (delays) as well as zero-error UART speed choices.

For example, on a standard 16MHz Arduino, achieving serial speeds using the standard 2x multiplying approach (i.e. 9600 baud and up) runs into trouble once speeds exceed 50kbit/s. Atmel datasheet tables for the 328P recommends the use of 250k, 500k, and 1Mbit/s speeds instead of the 230kbit/s that the IDE would logically point to at 16 or 20MHz processor speeds. As I recall, it has to do with how in-sync the CPU and the serial port are.

Reply

Lony
February 12, 2012 at 12:40 pm

I needed the 1284p for a project of mine but wanted to take things a little further so I mocked up a "ProMighty" using the SMD variant. It's a little larger than the ProMini but shorter than the DIP and 1 1/4 wider than the DIP package all in all a basic arduino in a slightly larger package then the DIP itself.

Thanks Manicbug!

Reply

maniacbug
February 12, 2012 at 6:11 pm

Cool! I can't wait to see it. Are you going to sell it? What pinout are you going to use?

Reply

Lony
February 12, 2012 at 7:41 pm

I'll send you a bare pcb when they arrive if your up to some soldering. The pins are arranged with port a and b on the right with vcc, gnd and reset to make accessing the pins easier for bootloading and c and d with raw (regulated) and gnd on the left, ftdi connects at the end of the board like the promini. I added pad for the reset switch on the bottom also to make it possiable to hit the reset no matter which side is up! 😊

Reply

Flo
February 13, 2012 at 12:54 am

I finally got to try it and it seems to work fine when using Arduino as ISP to program it but once i put a bootloader on it joy ends for me…
I dont have a FTDI cable so i'm using "Arduino USB2SERIAL Light" – which works fine with atmega328 and atmega168 chips (using optiboot).
But on the 1284p it just stops after (successfully) reading the device signature (out of sync).
I tried the original 16MHz bootloader and Optiboot, here is a verbose output from avrdude (Optiboot on the chip at that time):
http://pastebin.com/raw.php?i=rtwZ2sHQ

Maybe you ran across a similar problem while developing and could give me a hint or have any idea what i could try.
Thanks!

Reply

Lony
February 13, 2012 at 6:19 am

Seems as though you are not the only one having problems with USB2Serial light adapter see: http://arduino.cc/forum/index.php?topic=83215.0 I ,although not with the 1284p. Looks like the USB2Serial light adapter fails with other AVRs also, I hope this helps in some way.

Reply

maniacbug
February 13, 2012 at 12:29 pm

We're seeing someone else with this problem too, see http://arduino.cc/forum/index.php/topic,80483.msg688630.html#msg688630

Could you run again with 4 -v's instead of just 2? That will print the WHOLE verbose output. Pretty sure it'll look like like CrossRoads' results in that thread I linked to.

Does your USB2SERIAL also work when you're uploading to a 328p at 115200 baud? That is, are you running optiboot on a 328p and uploading to that successfully?

Reply

Flo
February 13, 2012 at 1:46 pm

Of course: http://pastebin.com/raw.php?i=9gbjTeck
I think i got a little bit further than CrossRoads…
Just tried lowering the Baud rate to 9600 and it worked 🙂
So lets see how fast we can go…

I'm using Optiboot on 328p with this cable – works fine.

Stephen
February 21, 2012 at 8:26 am

Hello,
Btw, great site! Now if I can only get it to work (i'm sure I have done something wrong).

First attempt at running a 1284p with optiboot. I followed the directions above and used my working 328p as an ISP. The bootloader uploaded fine and I got no errors, however when I try to upload the basic Blink sketch, I am getting an out of sync message in Arduino IDE. When the IDE uploads, the chip resets properly and the TX led on my FTDI flashes a couple of times, then I get the out of sync message.

When I press the reset button, the LED blinks 3 times, so in my mind that tells me the bootloader is there and working???

I have tried changing the baud rate from 115200 to 19200 and 9600. No luck.

Any suggestions would be appreciated.
-SK

Hardware: 1284p on a breadboard using an 8MHz crystal. FTDI USB to Serial chip on breadboard (works fine to my 328p on breadboard)
I recompiled Optiboot after changing the clock setting in Make file to 8000000L. I also changed the boards.txt file to support 8Mhz Optiboot on 1284p.
Here are the Make file settings and boards.txt settings that I used: http://pastebin.com/9dktvx2S

Reply

Stephen
February 22, 2012 at 9:13 am

Hello, figured it out (at least It is working now)

The baud rate was to fast for the 8MHz crystal. In Make file I changed -DBAUD_RATE=115200 to -DBAUD_RATE=19200, recompiled and made boards.txt match. Burned the new boot loader and my 1284p is now happily blinking away.

Tonight I'm going to up the baud rate to 57600 and see if it works at that rate as well. I'm also going to try other, faster crystals in the next couple of days to see if it can do 115200 at 16MHz and 20MHz.

Will update with what I find. Again, thank you, this site made getting the 1284p up and running a very easy process.
Stephen

Reply

maniacbug
February 22, 2012 at 7:26 am

You are not the only person who has had trouble uploading using a 1284 bootloader. I'm hopeful that someone will discover the breakthrough that makes it work for everyone else.

Can you try it with a 16MHz crystal? If it works in that case, we know it is directly a problem with 8Mhz.

Something else you can try, use the regular bootloader out of the box, but change the boards.txt baud rate to 57600. The bootloader is compiled for 115200, but running at 8Mhz will cut that in half.

Also try the "old" bootloader. There is a 8Mhz entry for that, and I have tested that one myself at 8MHz.

Good luck, and thanks for writing!

Reply

Steve Marple
February 22, 2012 at 7:32 am

Does uploading the bootloader 'preload' the blink sketch? Some do, and the very slow blinking helped me to figure out I had the fuses wrong and was using the internal oscillator and prescaler instead of the external crystal.

Reply

maniacbug
February 22, 2012 at 7:35 am

No, it doesn't. That's a good idea, though.

Alkex Smith
February 24, 2012 at 12:55 am

Hi,
I finally got my 1284P to play with but no joy uploading the bootloader using duemilanove as ISP (ATMEGA328P-PU).
I've spent hours checking the wires, re-started the breadboard 3 times. It is wired identically to your schematic, minus the FTDI connections (haven't received the FTDI gear yet)

I'm using the default ISP sketch that comes with Arduino IDE 1.0 – does this need a tweak of some kind?
Here is the error I'm getting:
http://pastebin.com/tScNj5Xj

And the markings on the chip (hope I got the right one!) ATMEGA1284P PU 1139

I'm considering getting an AVR programmer to rule out and possible probs with using Arduino as ISP – which would you recommend?

Thanks in advance for any suggestions!
Alex

Reply

maniacbug
February 26, 2012 at 7:31 am

Default Arduino ISP is flaky now, because the serial buffer changed to 64 bytes. The reliable fix is to change the serial buffer to >=100 bytes. Or try reducing the baud rate in the ISP sketch and the programmers.txt file.

I posted some about this on the forums:
http://arduino.cc/forum/index.php/topic,76138.msg676386.html#msg676386

Reply

Lony
February 26, 2012 at 12:30 pm

Here's a little more info on the ProMighty boards I had done up.

1) 500ma resetable fuse for USB on FTDI connection.
2) HC49US Landing pad for the crystal, drop in the crystal with the speed you need.
3) Landing pad for regulator and caps tied to RAW input pin.
4) FTDI connection on the end.
5) 2 rows of1 9 pins (one row on each side) breaking out all usable pins. Pin rows are spread 9 pins wide.
6) ISP pins are not brought out in 6 pin config but on one port/edge together inline. GND, VCC, RST, SCK, MISO, MOSI
7) Size is about the same as the DIP.
Link to Eagle export of .png http://img163.imageshack.us/img163/228/promighty.png

Reply

maniacbug
February 26, 2012 at 2:52 pm

Totally super hot. Love how small they are! Nice job. Thanks for sharing.

Reply

Lony
February 26, 2012 at 3:14 pm

I designed a MultiWii flight controller/shield around the ProMighty also. I'm waiting on testing the ProMighty on a perf board before submitting the flight controler/shield.

Sam….Urai!!!
March 13, 2012 at 11:54 pm

"Hours later I still cant figure this out. I have tried other bootloaders but with no luck."
Any help would be great!
Verbose: http://pastebin.com/9gG6xkSE

Reply

maniacbug
March 14, 2012 at 3:21 pm

You wounldn't be the first to have problems. It's not clear why it works for some and not for others. There is a thread on the Arduino forums you can try, other people are working through similar problems.

Reply

Sam….Urai!!!
March 14, 2012 at 4:46 pm

It seems people using the duemilanove have better luck… I have one on the way that should arrive tomorrow. If I have better results Ill share the fact.

Reply

Sam….Urai!!!
March 15, 2012 at 12:21 pm

Nope ☹

Reply

Sam….Urai!!!
March 15, 2012 at 1:10 pm

I decided to try and bootload it with windows, using a duemilanove and it worked. ☺ *Macbook Pro / OSX Lion / Intel Core i7 / 8BG ram* "Im a music/beat junkie" Defeat latency

Lony
March 14, 2012 at 6:09 pm

ProMighty bare boards.

Reply

Sam….Urai!!!
March 15, 2012 at 12:20 pm

Yeah I am planning to build a board similar to the ProMighty

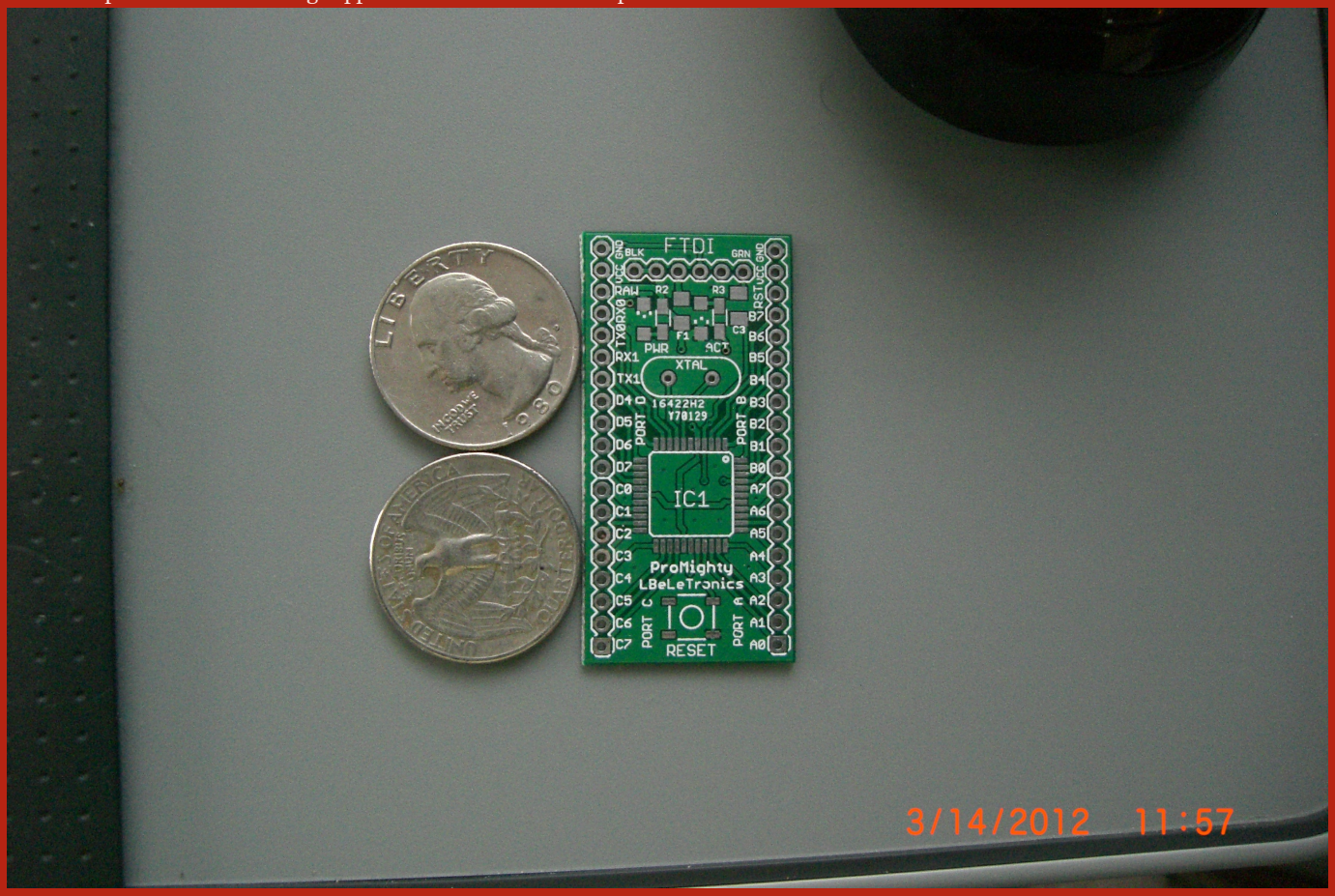
Reply

Sam….Urai!!!
March 15, 2012 at 12:30 pm

But we have a communication problem lol

Lony
March 15, 2012 at 1:25 pm

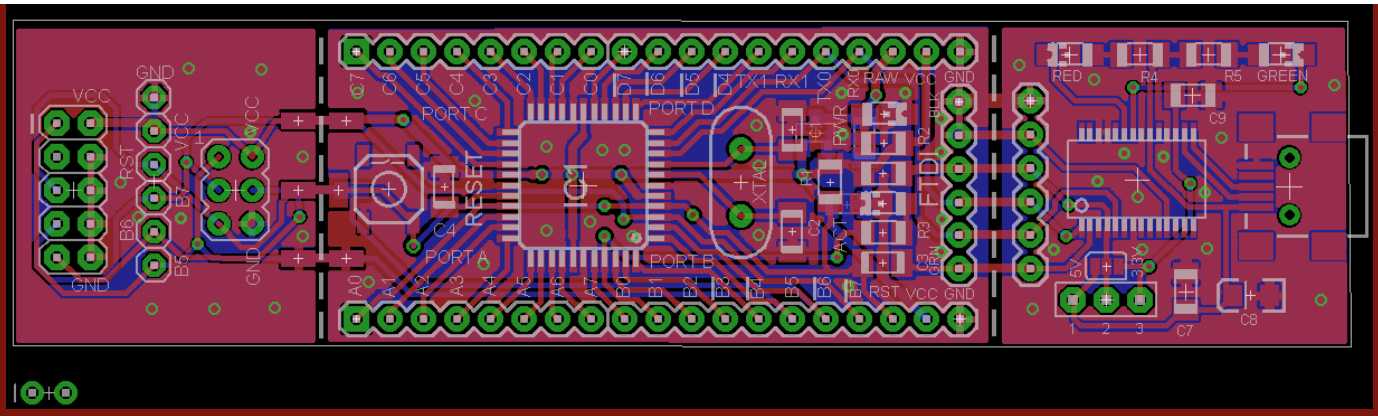
I recived the bare ProMighty boards I had fabricated but when I posted the “ProMighty bare boards” comment it was supposed to include a link to the pictures but something happened. Here’s another attempt at it



Reply

Lony
March 15, 2012 at 1:34 pm

Here is the All in one design! It should be here in a couple of weeks.



Reply

Sam….Urai!!!

March 15, 2012 at 2:40 pm

Awesome! Look good. This will be way more bad ass than the pro mini!

Reply

Lony

March 15, 2012 at 6:05 pm

First, I anyone is interested in the ProMighty bare boards $1.50 to US address includes shipping but I'll only ship once a week so I don't have to Post Office every day. The boards are SMD 1206 parts for easier soldering and I'll email a link to the project cart I have set up at Mouser.

Reply

Sam….Urai!!!

March 15, 2012 at 8:41 pm

Are you bootloading these with this bootloader? What ftdi are you using to write sketches? I am unable to write sketches to this using Sparkfuns FTDI basic / Uno / or Duemilanove…. Thanks

Reply

Lony

March 15, 2012 at 9:40 pm

My hand etched DIP prototype worked fine from what I tested. Still waiting on a few parts from Mouser. I had to reorder due to first shipment going to my Hawaii address but I'm in Alaska for spring.

theloone1

April 8, 2012 at 10:38 pm

OK! ProMighty is alive, well almost. I have a couple assembled along with having to make a 10 to 6 pin / ProMighty programing adapter for my USBAP, now the fun starts. 🙂 Find me here: http://arduino.cc/forum/index.php?topic=80483.270

Reply

scharkalvin

April 17, 2012 at 1:25 pm

Nice. I was thinking of porting the 1.0 arduino to the 1284P myself, glad to find this. One problem would be loading the bootloader from Arduino since the only programmers I have are a usbtiny and a jtagmkII. Of the two Arduino seems to only support the USBtiny which doesn't work above on 128K devices (actually it DOES! It just can't verify a download after the fact and errors out. There is a patch to avrdude that opt's out of the verify but that's a bit risky). However I added the following three lines to my Programmers.txt and now I can use my avrjtagmkII to burn boot loaders from arduion:

jtagmkII.name=AVR JTAGMKII
jtagmkII.communication=usb
jtagmkII.protocol=jtagmkII

In fact you can add ANY programmer supported by AVRdude this way!

Reply

David Garrison
April 20, 2012 at 12:26 pm

I'll take 3 of the ProMighty bare boards..email me the instructions on getting the funds to you

David Garrison
Gaithersburg MD

Reply

Captain Credible (@captaincredible)
May 5, 2012 at 10:53 am

Hi, where did you find the ATmega1284p that cheap ? At futurlec (where i usually find things cheapest) it costs $6.90

Reply

maniacbug
May 5, 2012 at 9:12 pm

That was the price on Mouser when I wrote it. Its gone up a little since then. Still well under $6.

Reply

scharkalvin
May 6, 2012 at 10:00 am

Mouser now has the DIP version for $5.82 in singles, $5.64 each for 10. Mouser usually has Atmel in singles for Digikey's 10 piece price and the two are the same at the 100 piece price. The QFTP parts are a few cents cheaper and are what I'd probably use if I lay out my own board. I found a nice 40pin DIP proto board from these guys (http://fun4diy.com/ AT40B) for $5.50. Connect a .1uf capacitor from pin 4 of the DB9 to the AVR reset line to be able to download sketches without hitting the reset button.

scharkalvin
May 6, 2012 at 10:11 am

BTW, When programming the boot loader onto the atmega1284p I would leave the JTAG fuse enabled (don't disable the JTAG) and modify the boot loader to disable the JTAG in software just before jumping to the user application. This way you can always connect the JTAG in AVR studio and connect by using the reset function if necessary and your application can use the I/O pins taken up by the JTAG.

Reply

Matt Williamson
May 10, 2012 at 6:09 am

I can't wait for my 1284s to get shipped. I'm so excited to try this out. The pin mapping text diagram is pretty mangled, at least in Chrome, could you add it as a .txt file in the Github repo?

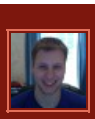
Reply

maniacbug
May 10, 2012 at 1:14 pm

It's in the pins_arduino.h file. https://github.com/maniacbug/mighty-1284p/blob/master/variants/standard/pins_arduino.h

Reply

Matt Williamson
May 29, 2012 at 8:20 am



Thanks. It looks great on this page now, too.

kuschelganxta
May 15, 2012 at 10:06 am

I had trouble with the ArduinoISP and the new chip. My USBtinyISP doesn't support >64kB flash.
Only this got me to get working: http://www.gammon.com.au/forum/?id=11637
Please, for all struggeling to get the bootloader on this chip, use the above link 🙂

Reply

scharkalvin
May 15, 2012 at 4:47 pm

The USB tiny might actually work with devices greater than 64k. It cannot verify such devices and you will get an error from Avrdude, but the device might actually have been correctly programmed.
I just bought myself an Atmel AvrispmkII for $35 and it works fine. However, the first time I try to program a virgin chip the fuse setting do not verify. They do the second time. I've been told that I need to select a slower ISP speed on virgin parts until the fuses have been set to use the external crystal.

Reply

mohiuddin
May 18, 2012 at 7:06 pm

please tell me how i use atmega8 instead of ft232. at the UNO and MEGA use

Reply

maniacbug
May 18, 2012 at 7:14 pm

Try posting on the Arduino forums.

Reply

Zhen
June 9, 2012 at 8:53 pm

Awesome tutorial on getting the 1284p to work.
However, I'm running into some problems getting both serial ports to work at the same time.
Any input to the first serial port resets me, as long as that serial port has been "begin()"
I can type a few characters at the IDE serial monitor and hit enter, and it will reset.
Serial1 works fine.
Anyone else have this problem? Or a solution?
Googling came up with nothing.
Another odd symptom is that if I just type a single character, it doesn't always reboot, but affects the output of Serial1.

Reply

maniacbug
June 10, 2012 at 2:44 pm

Never seen this before. Try posting to the Arduino forums.

Reply

Zhen
June 12, 2012 at 9:43 pm

Finally found my solution.
It wasn't easy trying to track it down, but here it is for anyone else having this problem:
http://www.avrfreaks.net/index.php?name=PNphpBB2&file=viewtopic&t=107115

scharkalvin
August 31, 2012 at 10:46 am

Most of the comments on this problem seem to indicate that the chip works fine at 16mhz, but fails at 20mhz. I'm not sure where exactly it DOES fail, but I'm going to back off from the 20mhz crystal I have soldered in now. Going down in frequency, 18.432mhz is a 'magic' number that divides evenly into most baud rates, so I ordered a few crystals on this frequency. Hopefully the chip will behave at this rate. If not, I'll use the 1k ohm / 100pf trick or back off to 16mhz … but I'd rather run it as close to 20mhz as possible to get close to a 5mhz SPI clock.

Berto Caroteno
June 11, 2012 at 6:49 pm

Can someone helpme with the Arduino 1.0.1 support to Atmega32 (same pinout, less flash) I dont know how modfy the cores in arduino beyond the boards.txt file. Thanks!!

Reply

maniacbug
June 11, 2012 at 8:09 pm

Try the Arduino forums. If SRAM is the ONLY difference, you may not need any changes to the core at all.

Reply

Jan van Bodegraven
June 25, 2012 at 3:23 am

Great stuff, my 1284p is happily blinking. Seams like the modifications you made in the core files have made it into the Arduino 1.0.1 release. Could you confirm that ?

Reply

maniacbug
July 17, 2012 at 2:51 pm

Yes, all the modifications I submitted made it into the core. There are still a set of changes for Bobuino I made which never got it, but for a simple breadboard setup, the shipped core is sufficient. That said, a 1284P 'variant' is still required which is not shipped with Arduino, so you still need that plus a boards.txt file.

Reply

Bajdi
July 14, 2012 at 3:35 pm

Thanks for the guide 🙂 I've just successfully bootloaded an ATmega1284P-PU. I used an Arduino Duemilanove as ISP and the Arduino 1.0.1 IDE, didn't encounter any problems.

Reply

Pito
July 15, 2012 at 3:46 am

Is the pin numbering on the analog pins correct (D24-D31)? P.

Reply

maniacbug
July 21, 2012 at 3:05 pm

Does something seem incorrect?

Reply

scharkalvin
July 26, 2012 at 12:18 pm

Now that Arduino 1.0.1 is out is the current release of maniacbug still compatible?

Reply

scharkalvin
July 31, 2012 at 1:41 pm

Looks like arduino 1.01 is compatible with your port, just need to move a few files around.

Just checked prices. Both DIgikey and Mouser are now $8.13 for a single piece, and $5.11 each for 25. Mouser will drop price to $6.79 each if you buy 10. Guess the law of supply and demand has come into play on this one. Glad I bought a few more to play with. Hate to say it but the xmega128a3u is about half the price at $4.75 (but no stock at the moment at digikey or mouser). Only half as much ram but more Usarts, more TWI and SPI I/O as well AND USB . Guess it's time to port arduino Leonardo to xmega!

Reply

John-Mike
August 2, 2012 at 10:26 am

To anyone having problems burning the bootloader with ArduinoISP, this worked for me
http://forums.reprap.org/read.php?13,119621,119802,quote=1

"
So I found that the reason for my problem with the Arduino as ISP was that any ArduinoISP sketch that is uploaded from the arduino 1.0 software doesn't work because of some sort of library mixup thing.

The solution was to upload the ArduinoISP sketch from an older version of the IDE, I used Arduino 0022 and it worked!
"

Reply

John-Mike
August 2, 2012 at 10:50 am

Hmm I spoke too soon. Out of sync when uploading sketches.

Reply

Alex Henrique
August 13, 2012 at 5:41 am

I want to use Atmega32. How it's possible?

Reply

maniacbug
August 18, 2012 at 6:39 am

I do not have experience with that MCU. Please try asking on the Arduino forums.

Reply

scharkalvin
September 12, 2012 at 10:27 am

I found this link: http://www.subtours.com/ralph/theory/atmega16witharduino.html which has a link to download files to run the mega32 on Arduino. I'm not sure exactly how to patch this into the Arduino source tree, but it should be similar to patching the maniacbug sources. Someone else has already done the work. Note that with just a few more #ifdef's sprinkled in you should be able to also get the mega162, mega8515 and mega8535 working as well.

scharkalvin
August 24, 2012 at 8:07 am

The atmega32 is mostly compatible with the atmega644p so it should be possible to recompile the optiboot bootloader and modify the boards.txt file to include it. Digital and analog pinouts are the same. Fuse settings may be compatible better check with the online avr fuse calculator though. The one area that may give problems is that some of the processor control register names and bit assignments have changed. Check with Atmel's document on migrating from the atmega32 to atmega324 to see what's different. You may have to edit some of the internal Arduino souce files to handle the differences (depending on what functions you are using). The atmega324p won't require these changes as it IS fully compatible with the atmega644p and atmega1284p register names and bit assignments

Reply

Michael LeBlanc
August 21, 2012 at 7:46 am

I'm wondering about the Rx/Tx connection between the FTDI and the 1284… in your schematic, you have Rx connected to Tx and vice versa. In my experience, it should be "like" pins connecting… shouldn't it?

Reply

maniacbug
August 22, 2012 at 8:55 pm

Nope, it's the opposite. One unit listens (RX) to what the other unit transmits (TX).

Reply

Bajdi
August 28, 2012 at 11:34 am

As I don't like putting big IC's on a breadboard I've designed a simple breakout board for the ATmega1284P-PU -> http://www.bajdi.com/bajduino-1284/ The pin numbering on the silkscreen uses the mighty 1284P layout.

Reply

Fab
September 26, 2012 at 5:30 pm

The only bootloader that works for me is that http://aka47.adsl24.co.uk/serendipity/index.php?/pages/Min644pWarez.html

Reply

kx
September 28, 2012 at 4:54 am

Hi, thanks for all the great work on making an arduino solution for the 1284P. I was just wondering, since they are all the same shape/part of the same family of chips, is it non-trivial to get this working on the atmega164 / 324 / 644 ? (particularly 164 in my case). Would it take a lot of effort to put together these versions? Of course I could help with testing for 164. Cheers

Reply

maniacbug
November 19, 2012 at 11:46 am

It seems like it should work on 164, etc. Please give it a whirl and let us know how it goes!

Reply

Sergi
October 9, 2012 at 5:59 am

Hi Maniacbug,

I will build an Arduino based on 1284p following your guide, however as it is my first time in arduino electronic design I will need some help. By the way I plan to do it with 20Mhz crystal, so I can take profit of the full chip, but I want it compatible with Arduino IDE and libraries. Some questions I have:
– Where can I find the bootloader for 20 Mhz?
– With 20Mhz, will the delay's and millis work propertly?

Thanks very much for your support, I would update on my progress.

Reply

maniacbug
November 19, 2012 at 2:26 pm

If it's your first time with electronic design, I recommend sticking with the tried-and-true formulations. In this case, I'd say stick with 16MHz. I do not have any information on 20Hz. Arduino forums may be some help, though.

Reply

scharkalvin
October 14, 2012 at 12:43 pm

I'm slowly going nuts here….
First my avrispmkII won't work at all with the atmega1284p, the bicolor light blinks orange an YET I've verified all the connections are correct. The programmer DOES work just fine with two different arduino boards. Also I can pull the processor out of the arduino and the avrisp STILL shows a green light but try that on my 40pin avr breadboard and the light STILL blinks orange.

I was able to burn and verify the optiboot bootloader on the '1284P using a JTAGMKII programmer (I changed the fuse settings to leave the JTAG enabled). I changed the clock frequency on the bootloader and in the boards.txt file to 18.432mhz which is an EXACT multiple of most common baud rates (Plus there has been some talk of problems with the atmega1284P at 20mhz so I decided to split the difference between 16mhz and 20mhz and cross my fingers). I plan on added two or three lines of code to the bootloader to disable the JTAG in software before jumping to the user program. That way I can keep the JTAG HW enabled.

HOWEVER…..
The bootstrap don't seem to work. I'm using an RS232 to serial chip (which SHOULD work at 115200 baud) and wired the DSR pin to reset via a cap but I still have to press the reset button to get things started downloading a sketch. Only it doesn't quite make it all the way….
(output of download command in Arduino ide….)

Binary sketch size: 3,192 bytes (of a 130,048 byte maximum)
/home/ken/arduino/arduino-1.0.1/hardware/tools/avrdude -C/home/ken/arduino/arduino-1.0.1/hardware/tools/avrdude.conf -v -v -v -v -patmega1284p -carduino -P/dev/ttyS0 -b115200 -D -Uflash:w:/tmp/build5445213971359224805.tmp/AnalogReadSerial.cpp.hex:i

avrdude: Version 5.11, compiled on Sep 9 2011 at 16:00:41
Copyright (c) 2000-2005 Brian Dean, http://www.bdmicro.com/
Copyright (c) 2007-2009 Joerg Wunsch

System wide configuration file is "/home/ken/arduino/arduino-1.0.1/hardware/tools/avrdude.conf"
User configuration file is "/home/ken/.avrduderc"
User configuration file does not exist or is not a regular file, skipping

Using Port : /dev/ttyS0
Using Programmer : arduino
Overriding Baud Rate : 115200
avrdude: Send: 0 [30] [20]
avrdude: Send: 0 [30] [20]

avrdude: Send: 0 [30] [20]
avrdude: Recv: . [14]
avrdude: Recv: . [10]
AVR Part : ATMEGA1284P
Chip Erase delay : 9000 us
PAGEL : PD7
BS2 : PA0
RESET disposition : dedicated
RETRY pulse : SCK
serial program mode : yes
parallel program mode : yes
Timeout : 200
StabDelay : 100
CmdexeDelay : 25
SyncLoops : 32
ByteDelay : 0
PollIndex : 3
PollValue : 0x53
Memory Detail :

Block Poll Page Polled
Memory Type Mode Delay Size Indx Paged Size Size #Pages MinW MaxW ReadBack
———— —— —— —— —— —— —— —— —— —— —— ——
eeprom 65 10 128 0 no 4096 8 0 9000 9000 0xff 0xff
Block Poll Page Polled
Memory Type Mode Delay Size Indx Paged Size Size #Pages MinW MaxW ReadBack
———— —— —— —— —— —— —— —— —— —— —— ——
flash 65 10 256 0 yes 131072 256 512 4500 4500 0xff 0xff
Block Poll Page Polled
Memory Type Mode Delay Size Indx Paged Size Size #Pages MinW MaxW ReadBack
———— —— —— —— —— —— —— —— —— —— —— ——
lock 0 0 0 0 no 1 0 0 9000 9000 0x00 0x00
Block Poll Page Polled
Memory Type Mode Delay Size Indx Paged Size Size #Pages MinW MaxW ReadBack
———— —— —— —— —— —— —— —— —— —— —— ——
lfuse 0 0 0 0 no 1 0 0 9000 9000 0x00 0x00
Block Poll Page Polled
Memory Type Mode Delay Size Indx Paged Size Size #Pages MinW MaxW ReadBack
———— —— —— —— —— —— —— —— —— —— —— ——
hfuse 0 0 0 0 no 1 0 0 9000 9000 0x00 0x00
Block Poll Page Polled
Memory Type Mode Delay Size Indx Paged Size Size #Pages MinW MaxW ReadBack
———— —— —— —— —— —— —— —— —— —— —— ——
efuse 0 0 0 0 no 1 0 0 9000 9000 0x00 0x00
Block Poll Page Polled
Memory Type Mode Delay Size Indx Paged Size Size #Pages MinW MaxW ReadBack
———— —— —— —— —— —— —— —— —— —— —— ——
signature 0 0 0 0 no 3 0 0 0 0 0x00 0x00
Block Poll Page Polled
Memory Type Mode Delay Size Indx Paged Size Size #Pages MinW MaxW ReadBack
———— —— —— —— —— —— —— —— —— —— —— ——
calibration 0 0 0 0 no 1 0 0 0 0 0x00 0x00

Programmer Type : Arduino
Description : Arduino
avrdude: Send: A [41] . [80] [20]
avrdude: Recv: . [14]
avrdude: Recv: . [03]
avrdude: Recv: . [10]
avrdude: Send: A [41] . [81] [20]

avrdude: Recv: . [14]
avrdude: Recv: . [04]
avrdude: Recv: . [10]
avrdude: Send: A [41] . [82] [20]
avrdude: Recv: . [14]
avrdude: Recv: . [05]
avrdude: Recv: . [10]
avrdude: Send: A [41] . [98] [20]
avrdude: Recv: . [14]
avrdude: Recv: . [03]
avrdude: Recv: . [10]
Hardware Version: 3
Firmware Version: 4.5
avrdude: Send: A [41] . [84] [20]
avrdude: Recv: . [14]
avrdude: Recv: . [03]
avrdude: Recv: . [10]
avrdude: Send: A [41] . [85] [20]
avrdude: Recv: . [14]
avrdude: Recv: . [03]
avrdude: Recv: . [10]
avrdude: Send: A [41] . [86] [20]
avrdude: Recv: . [14]
avrdude: Recv: . [03]
avrdude: Recv: . [10]
avrdude: Send: A [41] . [87] [20]
avrdude: Recv: . [14]
avrdude: Recv: . [03]
avrdude: Recv: . [10]
avrdude: Send: A [41] . [89] [20]
avrdude: Recv: . [14]
avrdude: Recv: . [03]
avrdude: Recv: . [10]
Vtarget : 0.3 V
Varef : 0.3 V
Oscillator : 28.800 kHz
SCK period : 3.3 us

avrdude: Send: A [41] . [81] [20]
avrdude: Recv: . [14]
avrdude: Recv: . [04]
avrdude: Recv: . [10]
avrdude: Send: A [41] . [82] [20]
avrdude: Recv: . [14]
avrdude: Recv: . [05]
avrdude: Recv: . [10]
avrdude: Send: B [42] . [82] . [00] . [00] . [01] . [01] . [01] . [01] . [03] . [ff] . [ff] . [ff] . [ff] . [01] . [00] . [10] . [00] . [00] . [02] . [00] . [00] [20]
avrdude: Recv: . [14]
avrdude: Recv: . [10]
avrdude: Send: E [45] . [05] . [08] . [d7] . [a0] . [00] [20]
avrdude: Recv: . [14]
avrdude: Recv: . [10]
avrdude: Send: P [50] [20]
avrdude: Recv: . [14]
avrdude: Recv: . [10]
avrdude: AVR device initialized and ready to accept instructions

Reading | avrdude: Send: u [75] [20]
avrdude: Recv: . [14] . [1e] . [97] . [05] . [10]
| 100% 0.00s

avrdude: Device signature = 0x1e9705
avrdude: Send: V [56] . [a0] . [0f] . [fc] . [00] [20]
avrdude: Recv: . [14]
avrdude: Recv: . [00]
avrdude: Recv: . [10]
avrdude: Send: V [56] . [a0] . [0f] . [fd] . [00] [20]
avrdude: Recv: . [14]
avrdude: Recv: . [00]
avrdude: Recv: . [10]
avrdude: Send: V [56] . [a0] . [0f] . [fe] . [00] [20]
avrdude: Recv: . [14]
avrdude: Recv: . [00]
avrdude: Recv: . [10]
avrdude: Send: V [56] . [a0] . [0f] . [ff] . [00] [20]
avrdude: Recv: . [14]
avrdude: Recv: . [00]
avrdude: Recv: . [10]
avrdude: reading input file "/tmp/build5445213971359224805.tmp/AnalogReadSerial.cpp.hex"
avrdude: writing flash (3192 bytes):

Writing | avrdude: Send: U [55] . [00] . [00] [20]
avrdude: Recv: . [14]
avrdude: Recv: . [10]
avrdude: Send: d [64] . [01] . [00] F [46] . [0c] . [94] G [47] . [00] . [0c] . [94] r [72] . [00] . [0c] . [94] r [72] . [00] . [0c] . [94] r [72] . [00] . [0c] . [94] r [72] . [00] . [0c] . [94] r [72] . [00] . [0c] . [94] r [72] . [00] . [0c] . [94] r [72] . [00] . [0c] . [94] r [72] . [00] . [0c] . [94] r [72] . [00] . [0c] . [94] r [72] . [00] . [0c] . [94] r [72] . [00] . [0c] . [94] r [72] . [00] . [0c] . [94] r [72] . [00] . [0c] . [94] r [72] . [00] . [0c] . [94] r [72] . [00] . [0c] . [94] r [72] . [00] . [0c] . [94] r [72] . [00] . [0c] . [94] . [b0] . [00] . [0c] . [94] r [72] . [00] . [0c] . [94] . [b7] . [01] . [0c] . [94] M [4d] . [02] . [0c] . [94] r [72] . [00] . [0c] . [94] r [72] . [00] . [0c] . [94] r [72] . [00] . [0c] . [94] r [72] . [00] . [0c] . [94] r [72] . [00] . [0c] . [94] r [72] . [00] . [0c] . [94] . [f0] . [01] . [0c] . [94] . [9a] . [02] . [0c] . [94] r [72] . [00] . [0c] . [94] r [72] . [00] . [0c] . [94] r [72] . [00] . [0c] . [94] r [72] . [00] . [0c] . [94] r [72] . [00] . [0c] . [04] . [11] $ [24] . [1f] . [be] . [cf] . [ef] . [d0] . [e4] . [de] . [bf] . [cd] . [bf] . [11] . [e0] . [a0] . [e0] . [b1] . [e0] . [e6] . [e6] . [fc] . [e0] . [00] . [e0] . [0b] . [bf] . [02] . [c0] . [07] . [90] . [0d] . [92] . [a2] 1 [31] . [b1] . [07] . [d9] . [f7] . [1b] . [be] . [12] . [e0] . [a2] . [e1] . [b1] . [e0] . [01] . [c0] . [1d] . [92] . [a9] 6 [36] . [b1] . [07] . [e1] . [f7] . [10] . [e0] . [ce] . [e8] . [d0] . [e0] . [04] . [c0] " [22] . [97] . [fe] . [01] . [0e] . [94] – [2d] . [06] . [cc] 8 [38] . [d1] . [07] . [c9] . [f7] . [0e] . [94] . [a7] . [01] . [0c] . [94] 1 [31] . [06] . [0c] . [94] . [00] . [00] . [88] . [e1] . [0e] . [94] . [8e] . [00] . [bc] . [01] . [8b] . [e2] . [92] . [e0] J [4a] . [e0] P [50] . [e0] . [0e] . [94] . [a4] . [05] a [61] . [e0] p [70] . [e0] [20]
avrdude: ser_recv(): programmer is not responding
avrdude: stk500_recv(): programmer is not responding

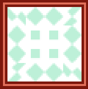
RATS!
Any ideas?

Reply

scharkalvin
October 19, 2012 at 2:06 pm

Problems solved. For some reason I could not use a 100nf cap between DTR and Reset with the '1284P. I had to put a darlington transistor between the reset pin and the DTR signal to get enough drive for the reset line on the processor. I used two transistors in a MMPQ2222 darlington connected. 10k resistor from base to ground, 100nf from DTR to base, collectors to Reset pin. I can now upload sketches without pressing reset.

Reply



Cam
October 14, 2012 at 12:50 pm

I have had success with at USBtinyISP – chucks up a verify error at the end but the booloaders does make it through.
I settled on the Pololu USB AVR Programmer which works just great for me & no errors.

Reply

maniacbug

November 19, 2012 at 11:42 am

Glad to hear it!

Reply

Going Bust
October 17, 2012 at 2:48 pm

The datasheet says six PWM Channels. Then it identifies the PWM able pins as:

SCK/OC3B/PCINT15 – Port B, Bit 7
MISO/OC3A/PCINT14 – Port B, Bit 6
SS/OC0B/PCINT12 – Port B, Bit 4
AIN1/OC0A/PCINT11, Port B, Bit 3
OC2A/PCINT31 – Port D, Bit 7
ICP1/OC2B/PCINT30 – Port D, Bit 6
OC1A/PCINT29 – Port D, Bit 5
OC1B/XCK1/PCINT28 – Port D, Bit 4

So I wonder if you can set up at most 6 of these as PWM pins or what. It's a little hard to understand. Maybe I should just try running all six at once and see what happens.

Reply

maniacbug
November 19, 2012 at 2:05 pm

This would be a great question for the Arduino forums!

Reply

Steve Marple
November 19, 2012 at 2:16 pm

You can if you aren't using the timer for anything else. The datasheet is a bit vague (or misleading) on which MCUs have 6 PWM outputs. I only made it work with the ATmega1284P. If you follow the 'standard' Arduino pin mapping and use SCLK for D13 then you'll notice that this is where the built-in LED is normally connected 🙂 Instead of the standard blink script you can easily fade the LED up and down in brightness. This is the test sketch I normally run after I've built a new Calunium board.

Reply

James
October 23, 2012 at 3:03 am

Does anyone know of a bootloader for the 1284P-AU (smt) version? Ideally looking for 20Mhz also… Thanks.

Reply

maniacbug
November 19, 2012 at 11:58 am

I don't think there's a difference for surface-mount version. Others have used SMD chips with these boothloaders plenty.

Reply

scharkalvin
November 19, 2012 at 5:40 pm

The optiboot loader works fine with my ATmega1284p. I am running mine at 18.432mhz (this frequency is a perfect multiple of the standard baud rates). I had a small issue with my processor however. Connecting a .1uf capacitor to DTR to allow avrdude to reset the processor did not work. A scope confirmed that the line was NOT being pulled down. It seems that at least some of the early versions of this processor have a VERY strong internal pullup on the reset line! I ended up putting two darlington connected 2n2222 transistors between

the capacitor and the reset line. The base of the transistor goes to ground via a 10k resistor and to the DTR line via a .luf cap. The collector(s) go to the reset line. I used a quad 2n2222 package with two of the transistors unused.

Reply

scharkalvin
November 19, 2012 at 5:48 pm

You'll have to recompile from source for 20mhz (or any other clock frequency). Just edit the makefile, no problem.

Reply

Preston
January 4, 2013 at 2:03 am

Dumb question – Why is an external oscillator necessary? Digikey lists this chip as having an internal 20MHz oscillator. -Thanks

Reply

maniacbug
July 8, 2013 at 6:34 am

Arduino forums have more experts in this area. My understanding is that the internal oscillator isn't sufficiently precise for serial communication.

Reply

Pingback: "Goldilocks" 1284p Arduino UNO Clone | feilipu

feilipu
March 8, 2013 at 10:32 pm

I'm getting some professional Arduino Uno clones made, with 1284p processors. 100% Shield compatibility, with 8 times the SRAM is the justification. It is the Goldilocks solution; not too small (insufficient SRAM), not too big (Arduino Mega Shield incompatibilities). Constructor will be Freetronics in Australia. http://pozible.com/goldilocks You can get them over-clocked to 22.1184MHz, if you want perfect serial synchronisation, or just extra 38% speed. Please grab some.

Reply

xarolium
April 15, 2013 at 11:54 am

Dear Diyer,

I upload the Mighty bootloader on a Atmega 1284p without any problem.
But now i've some trouble when i tr to upload my sketch with the Mini USB/serial adapter.
I connect two 1k resistor in serie with RX and TX but uploading work like once in ten. Is there something to do to solve this problem?
thank ou for your help,

Xarolium

Reply

Jim Mchugh
April 30, 2013 at 10:20 am

Maniacbug!

Great work!

I'm going to try to combine two of your projects.

I'm going to build a quadcopter controller using an Atmel ATmega1284P and your RF24 Network stack.

One question:

Does your version of the ATmega1284P Arduino clone support SPI fully?

I will obviously need that to talk to my nRF2401 module.

Cheers!

Reply

maniacbug
July 8, 2013 at 6:14 am

Sounds awesome, I hope this went well!

Reply

maniacbug
July 11, 2013 at 7:34 pm

Sounds great. Good luck with your project! The SPI works great, I use it all the time.

Reply

3dotter
May 21, 2013 at 8:37 am

I also need a proper bootloader for the ATMEL – ATMEGA1284P-AU – MCU, 8BIT, AVR, 128K FLASH, 44TQFP smt version. Which one is best and can this MCU run with this bootloader at 16 MHz as well at 3.3 Volt?

I am making an Arduino GPS GSM wildlife tracker with this MCU. The PCB's have arrived already from OSHPark, the purple PCB Provider. They look great!

Thanks in advance for your answers. 🙂

Reply

maniacbug
July 11, 2013 at 7:29 pm

Not sure about the best bootloader for this particular chip. You could try asking on the Arduino forums. Good luck!

Reply

Sam Weiss
June 7, 2013 at 4:43 pm

Hi

After a a few trials and tribulations of learning how to burn bootloaders (and learning to double check, and re-doublecheck wiriing), I managed to get the bootloader onto the mega1284P, and get a basic "blink" sketch going.

What I have found, though, is that the autoreset from the ftdi interface isn't working, and I have Pin 9 tied down to +5 with a 10k resistor, and is also attached to dtr though a 0.1uf cap (as shown above in the schematic). I'm using the sparkfun ftdi chip.

What I found is some conversation from a link on their atmega328 chip about autoreset and optiboot. It basically said that sometimes the autoreset won't work if the baudrate is 115200. I'm wondering how I can knock it down to 57600. Does editing the boards.txt, and re-uploading the bootloader set it to 57600, or do I need to re-compile optiboot? Where is the baud rate setting in optiboot? Or can I use a different size capacitor between dtr and autoreset. I've already tried several different values, but only what I had on hand.
https://code.google.com/p/optiboot/issues/detail?id=28#makechanges

it's possible that a new version of optiboot has fixed this problem, but I'm not sure what the date is of the optiboot in your package, which is fine by me, so long as I can get it to work. I think I can wait the extra seconds it will take for things to upload at the same rate.

Thanks!

Sam

Reply

Sam Weiss
June 7, 2013 at 4:47 pm

Btw, I really like the work you have done. Not only was this article really great in giving me a possible solution to my impending space (and io pin shortage), I'm also using your RF24 library, and it was the only library that had the features I needed for NRF24L01.

Reply

Stewart
July 8, 2013 at 9:24 pm

Hi,
Inspirational work, I found it most helpful.
I have just built my 1st Arduino programmable ATmega1284 board and am currently testing. So far, so good.
Its minimal footprint and once I'm happy with its behavior I'll release the PCB files and other relevant data (PCB in Kicad).
It fits in the minimum footprint for Seedstudio (its 49.5mm x 32mm)
Extra hardware on board is Wiz820 ethernet, micro-SD card, DS1307 batter-backed RTC and a RS485 transceiver on the 2nd serial port.
You can see pictures at:
http://byremote.blogspot.com.au/2013/07/the-new-mini-biggie.html

Reply

maniacbug
July 11, 2013 at 7:10 pm

Looks great–nicely done!

Reply

konfu
August 9, 2013 at 6:39 am

I just tried to use the 1284P and added a software serial on pins A6 (RX) and A5 (TX). Seems like this doesn't work at all. Damn. Did anyone ever tried software Serial with the 1284P on Arduino?

Reply

konfu
August 9, 2013 at 7:29 am

Just found the solution. It's about the pinchange interrupt not being implemented correctly. I've used the code shown in this thread and it just works great:

http://forum.arduino.cc/index.php?topic=157297.0

Reply

feilipu
August 18, 2013 at 4:18 am

Following up on my comment from March. The Goldilocks 1284p project is now finished, and the Pozible Pledges are shipping out tomorrow.
http://www.pozible.com/goldilocks

Freetronics (who helped me to realise the board) are going to be selling them too, and they're available soon in their web site http://www.freetronics.com/goldilocks.

We used the stk500v2 bootloader as the starting point for the 1284p, and the current LUFA code for DFU and USB to Serial (U2duino) wiring compatible on the 32u2. As soon as the Pozible shipments are out, then we'll put the (Creative Commons) design files up for reference. The code set is available here: http://feilipu.me/2013/03/08/goldilocks-1284p-arduino-uno-clone/

Thanks again for the inspiration.

Reply

feilipu
August 18, 2013 at 4:20 am

Oops. I forgot to mention there's a User Manual http://tinyurl.com/Goldilocks1284p

Reply

jason
August 26, 2013 at 2:17 pm

is there anyway to make the pwm frequency library to work on a 1284p. it compiles fine on the mega 256 but not on the 1284p. i tried altering the library files so the 1284 is in the susported list but thats as far as i could do as im no programmer. this was suggested on another site but didnt work, i still get the sam error

Reply

Dominic
September 5, 2013 at 12:07 pm

Is this not updated to allow for INPUT_PULLUP on pinMode? how do we define the pullup method for pinMode on the mighty1284p?

Reply

maniacbug
October 5, 2013 at 7:06 am

Indeed, the mighty 1284p core is a little old now, so recent changes to the Arduino core aren't always reflected. This sounds like one of them.

Reply

Kevin Dethlefs-Moreno
September 20, 2013 at 1:26 pm

Greetings,

Excellent work here.

I'm having a hard time finding contact information for you. I'm reaching out to you to ask some basic questions regarding the licensensing of your code on GitHub. A document is not provided and thus no license is specified, not permitting me to make any assumptions.

I would like to include parts of your source code (primarily Arduino.h for the 1284 modified minimally for the ATMega324P chip) in a custom board definition for a robotic project. We will want to release this integration into Arduino to the public with a license of our choosing, but need to know what restrictions you include.

Please contact me at kdethlefsmoreno@unomaha.edu.

Thank you,
Kevin Dethlefs-Moreno

Reply

maniacbug
October 3, 2013 at 6:57 am

Hi, the 1284P platform is a derivative of the Arduino code, so licensed the same as that. I think they use MIT license.

Reply

phenyl

October 1, 2013 at 4:32 am

Hi,
Thank you very much for your work. I just bought an atmega1284p and installed your “Mighty1284p 16 MHz using Optiboot” bootloader on it. Unfortunately, I can’t seem to get an SD card working with it. I have a bigg-ish code which works if I comment out all sd-related stuff. SD-Breakout using SCK, MOSI, MISO and CS (pin 5, SS, D4).

Have you maybe come across this problem before?

```
...
#define CS 4
...
setup
...
    Serial.print("Initializing SD card...");
  pinMode(CS, OUTPUT);
  digitalWrite(CS, HIGH);
  if (!SD.begin(CS)) {
      Serial.println("Card failed, or not present");
    return;
  }
    Serial.println("Card initialized.");
...
```

It seems to be hanging at the initialization of the SD-card.

Thank you very much for any hint.

Reply

phenyl
October 1, 2013 at 4:33 am

(The whole code works perfectly on an arduino Mega 2560, but as I want to put it somewhere permanently without locking away an entire arduino…)

Reply

phenyl
October 1, 2013 at 5:21 am

I think I got it working.
in
/Applications/Arduino.app/Contents/Resources/Java/libraries/SD/utility/Sd2PinMap.h
one needs to add to the atmega644 line so that it reads:

```
#elif defined(  AVR ATmega644P  ) || defined(  AVR ATmega644  )|| defined(  AVR ATmega1284P  )
```

Then my code compiled and it now seems to run without a hitch.

Stewart
October 2, 2013 at 8:47 pm

1284 board files are now available (board pics at http://byremote.blogspot.com.au/2013/08/mini-web-atmega1284p-board-v2.html)
Files at: http://www.pokewithastick.com, these are in Kicad format. PCB is 49.5 x 37mm
They PCB’s are cheap to get made but the SM components may put some people off.
I have been programming it in Arduino as libraries are readily available for the peripheral hardware.

When fully populated you have:
Ethernet (Wiz820)
micro SD card
battery-backed RTC
2 x logic level serial or 1 x logic level and 1 RS485
connector for nRF24L01 radio
user LED
power LED
reset button.

Available pins are almost all on a 0.1″ matrix so you can plug in onto stripboard.
6-pin Atmel ISP header for programming or by Arduino serial (FDTI basic lead) if a suitable bootloader installed (I used the Goldilocks one for testing). Serial programming needs a harness as the pin layout does not match.

If its useful to anyone, enjoy 🙂
Stewart

Reply

scharkalvin
December 14, 2013 at 8:24 pm

This has been mentioned before elsewhere, but as some have found out there are two versions of the atmega1284, one is the atmega1284P-PU, the other is the atmega1284-PU. They have different signatures, and the NON "P" part ISN'T supported by the version of GCC and avrdude supplied with arduino. HOWEVER, the parts are software identical. I was able to burn the same bootloader (optiboot) in both parts without recompiling (same hex file) but I had to use avrdude from the command line with the -F parameter to force the tool to ignore the signature. I've made two changes to the optiboot source. First, I'm running my processor at 18.432 mhz. Second, I do NOT disable the jtag with the fuse settings, rather I have the bootloader disable the jtag in software (writting to the control register) just before jumping to the user code. This way I can STILL use the jtag if I want to without having to change the fuses. (Of course if you DO use the jtag you will have to replace the bootloader when you are done).

Reply

Peter Scargill
February 7, 2014 at 3:47 pm

Is it just me… I've just downloaded Arduino 1.55 – ie the latest… and of course the 1284 board isn't in there. I've tried copying the mighty-1284p directory as before … and the board is still not showing up… am I missing something? Bug in the beta or has anyone had success?

Reply

feilipu
October 26, 2014 at 3:42 am

Testing of my new Goldilocks Analogue prototype, including MCP4822.

Reply

feilipu
October 26, 2014 at 3:50 am

Oops. Swallowed the URL.
http://feilipu.me/2014/10/26/goldilocks-analogue-part-4/

Reply

Leave a Reply

Enter your comment here...