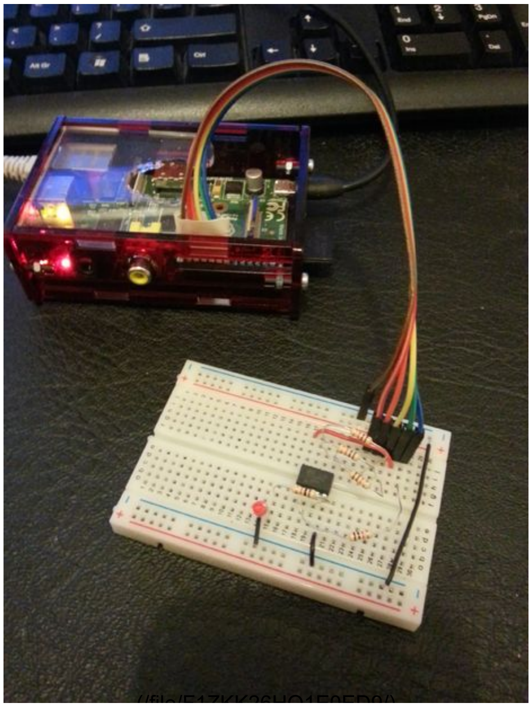
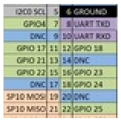


(/file/F17KK2SHO1F0FD9/)

About This Instructable

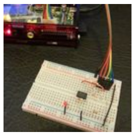
72,829 views
102 favorites

License: BY-NC-SA



prb3333 (/member/prb3333/)
Follow 18
(/member/prb3333/)

More by prb3333:



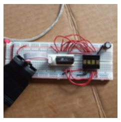
(/id/Programming-the-ATtiny85-from-Raspberry-Pi/)

(/id/Raspberry-Pi-Bluetooth-InOut-Board-or-Whos-Hom/)

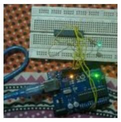
(/id/Raspberry-Pi-Snail-Mail-System/)

Related

Help: An Absolute Beginner's Guide to 8-Bit AVR Programming-AVR Dragon (/id/Help%3a-An-Absolute-

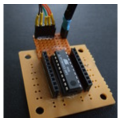


Ghetto Development Environment (/id/Ghetto-Development-Environment/)
by The Real Elliot



Programming AVR with Arduino as ISP without bootloader and external crystal (/id/Programming-

Getting started with ubuntu and the AVR dragon (/id/Getting-started-with-ubuntu-and-the-AVR-dragon/)



Ghetto Programming: Getting started with AVR microprocessors on the cheap. (/id/Ghetto-

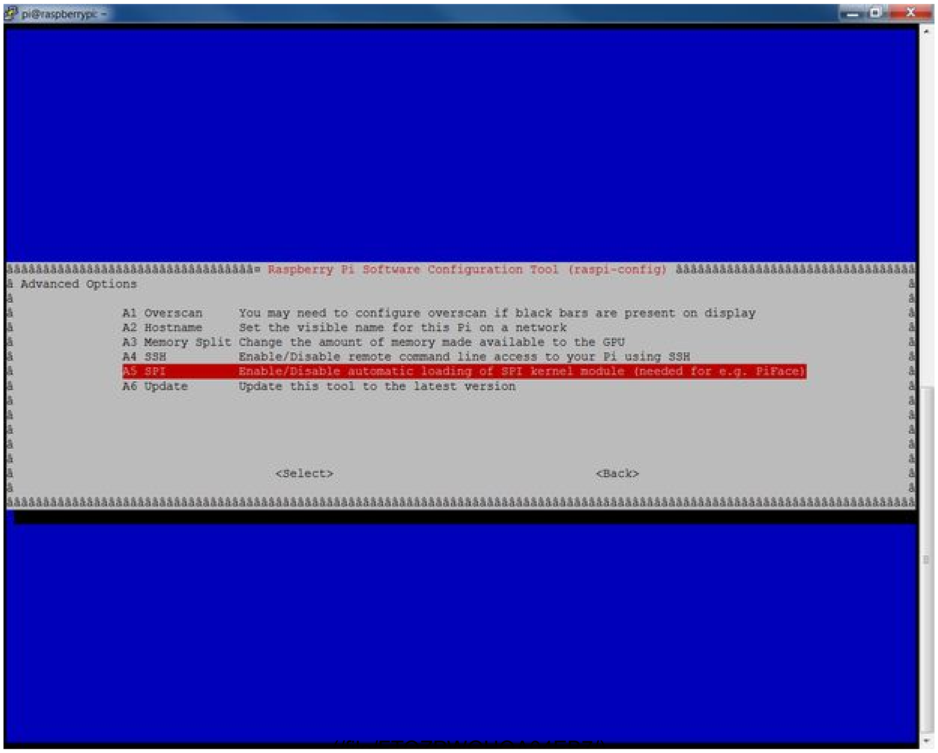
These instructions tell you how to setup and program the ATtiny85 microcontroller from a Raspberry Pi via the SPI interface. Lots of people use the Ardiuno to do this (then you can use the Arduino IDE and simplified C commands), or you can use a USB based programmer. I do not have an Ardiuno and don't want to buy a dedicated programmer. I do have a Pi, so I was pleased to learn I could use it as a way to get into microcontroller programming.

You will need:
Raspiberry Pi
ATtiny85 chip
5 x 1K resistors (or similar)
LED of your choice
A connection to the GPIO of the Pi, and a breadboard and wire.

Based on https://www.instructables.com/id/How-to-program-ATt... (https://www.instructables.com/id/How-to-program-ATtiny85-with-Raspberry-Pi/) and http://www.raspberrypi.org/phpBB3/viewtopic.php?t=... (http://www.raspberrypi.org/phpBB3/viewtopic.php?t=22650&p=317540) and https://www.instructables.com/id/RGB-LED-Mood-Light... https://www.instructables.com/id/ATTiny4585-LCD-display-control-with-a-shift-regis/step3/Programming-the-ATTiny85-with-a-test-program/

(https://www.instructables.com/id/RGB-LED-Mood-Light-w-ATtiny85/)

Step 1: Setup the Raspberry Pi



```
ââââââââââââââââââââââââââââââââââ¤ Raspberry Pi Software Configuration Tool (raspi-config) ââââââââââââââââââââââââââââââââââ
â Advanced Options                                                                                                          â
â                                                                                                                           â
â               A1 Overscan     You may need to configure overscan if black bars are present on display                     â
â               A2 Hostname     Set the visible name for this Pi on a network                                               â
â               A3 Memory Split Change the amount of memory made available to the GPU                                       â
â               A4 SSH          Enable/Disable remote command line access to your Pi using SSH                              â
â               A5 SPI          Enable/Disable automatic loading of SPI kernel module (needed for e.g. PiFace)              â
â               A6 Update       Update this tool to the latest version                                                      â
â                                                                                                                           â
â                                                                                                                           â
â                                                                                                                           â
â                                                                                                                           â
â                                                                                                                           â
â                          <Select>                                          <Back>                                         â
â                                                                                                                           â
ââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââ
```

At the terminal of the Pi:

Download and build avrdude

sudo apt-get install bison automake autoconf flex git gcc
sudo apt-get install gcc-avr binutils-avr avr-libc
git clone https://github.com/kcuzner/avrdude
(https://github.com/kcuzner/avrdude)
cd avrdude/avrdude
./bootstrap && ./configure && sudo make install

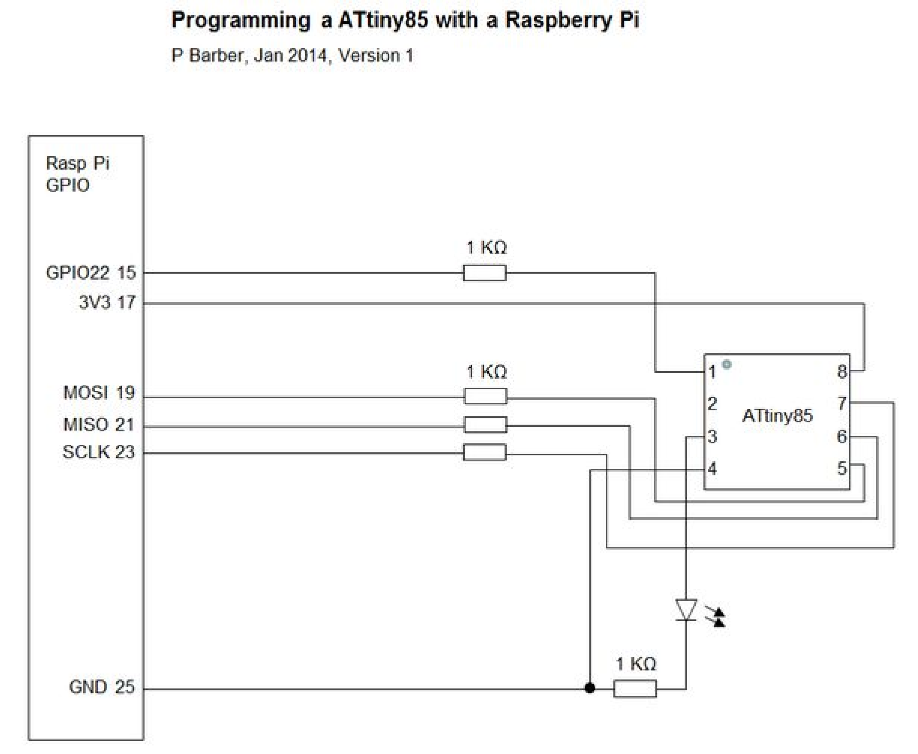
Setup SPI on the GPIO

sudo raspi-config
and Enable SPI device in the Advanced Options (see picture)
You can check this at the command line with lsmod, no need to reboot. (Maybe need to sudo modprobe spidev)

Download and build WiringPi for the gpio commands

cd ~
git clone git://git.drogon.net/wiringPi
cd wiringPi
./build

Step 2: Electrical Connections



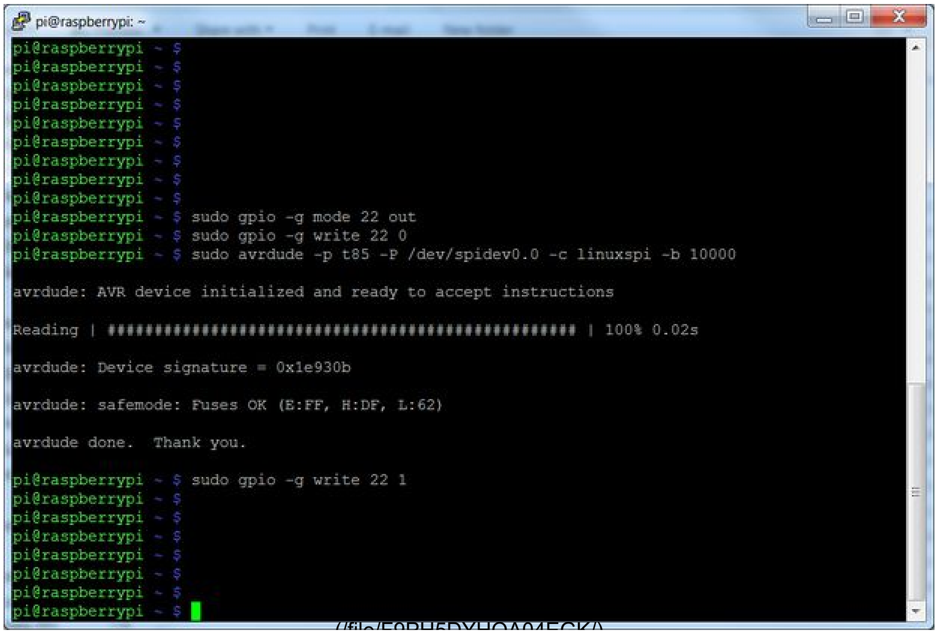
(/file/FX1LMLDHOA04EQB/)

Connect up the ATtiny85 to the Raspberry Pi GPIO (wire colours from the picture are given for reference):

GPIO pin	ATtiny pin	Comment
15	1	GPIO22 to Reset (through 1K, Blue wire)
17	8	3.3 V (Green wire)
19	5	MOSI (through 1K, Yellow wire)
21	6	MISO (through 1K, Orange wire)
23	7	SCLK (through 1K, Red wire)
25	4	GND (Brown wire)

(I could not find a way to do a nice table in instructables)

Step 3: Test Avrdude Connection



pi@raspberrypi: ~

```
pi@raspberrypi ~ $
pi@raspberrypi ~ $
pi@raspberrypi ~ $
pi@raspberrypi ~ $
pi@raspberrypi ~ $
pi@raspberrypi ~ $
pi@raspberrypi ~ $
pi@raspberrypi ~ $
pi@raspberrypi ~ $
pi@raspberrypi ~ $ sudo gpio -g mode 22 out
pi@raspberrypi ~ $ sudo gpio -g write 22 0
pi@raspberrypi ~ $ sudo avrdude -p t85 -P /dev/spidev0.0 -c linuxspi -b 10000

avrdude: AVR device initialized and ready to accept instructions

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

avrdude: Device signature = 0x1e930b

avrdude: safemode: Fuses OK (E:FF, H:DF, L:62)

avrdude done.  Thank you.

pi@raspberrypi ~ $ sudo gpio -g write 22 1
pi@raspberrypi ~ $
pi@raspberrypi ~ $
pi@raspberrypi ~ $
pi@raspberrypi ~ $
pi@raspberrypi ~ $
pi@raspberrypi ~ $
pi@raspberrypi ~ $
pi@raspberrypi ~ $
```

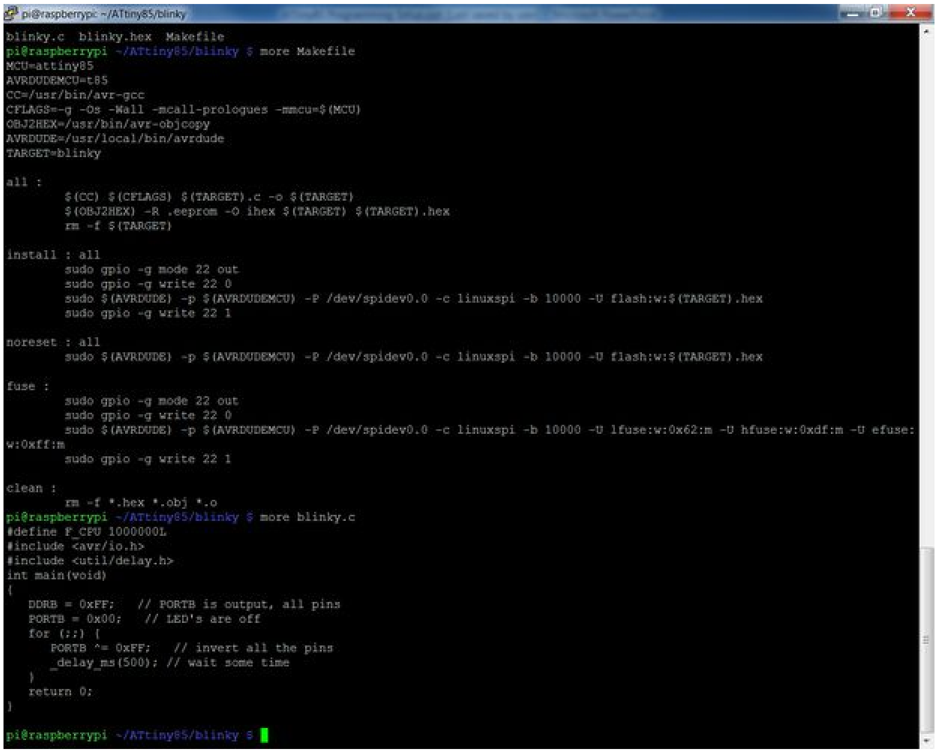
(/file/F9RH5DYHOA04FCK/)

Test avrdude connection to the ATtiny85, we are set up with GPIO pin 22 on the ATtiny reset. We must pull this pin low to program the chip. This can be done in other ways, e.g. a switch, but I an using another pin of the GPIO to do this.

```
sudo gpio -g mode 22 out
sudo gpio -g write 22 0
sudo avrdude -p t85 -P /dev/spidev0.0 -c linuxspi -b 10000
sudo gpio -g write 22 1
```

This must give success type messages!

Step 4: Program the ATtiny85



```
blinky.c  blinky.hex  Makefile
pi@raspberrypi ~/ATtiny85/blinky $ more Makefile
MCU=attiny85
AVRDUDEMCU=t85
CC=/usr/bin/avr-gcc
CFLAGS=-g -Os -Wall -mcall-prologues -mmcu=$(MCU)
OBJ2HEX=/usr/bin/avr-objcopy
AVRDUDE=/usr/local/bin/avrdude
TARGET=blinky

all :
        $(CC) $(CFLAGS) $(TARGET).c -o $(TARGET)
        $(OBJ2HEX) -R .eeprom -O ihex $(TARGET) $(TARGET).hex
        rm -f $(TARGET)

install : all
        sudo gpio -g mode 22 out
        sudo gpio -g write 22 0
        sudo $(AVRDUDE) -p $(AVRDUDEMCU) -P /dev/spidev0.0 -c linuxspi -b 10000 -U flash:w:$(TARGET).hex
        sudo gpio -g write 22 1

noreset : all
        sudo $(AVRDUDE) -p $(AVRDUDEMCU) -P /dev/spidev0.0 -c linuxspi -b 10000 -U flash:w:$(TARGET).hex

fuse :
        sudo gpio -g mode 22 out
        sudo gpio -g write 22 0
        sudo $(AVRDUDE) -p $(AVRDUDEMCU) -P /dev/spidev0.0 -c linuxspi -b 10000 -U lfuse:w:0x62:m -U hfuse:w:0xdf:m -U efuse:
w:0xff:m
        sudo gpio -g write 22 1

clean :
        rm -f *.hex *.obj *.o
pi@raspberrypi ~/ATtiny85/blinky $ more blinky.c
#define F_CPU 1000000L
#include <avr/io.h>
#include <util/delay.h>
int main(void)
{
   DDRB = 0xFF;   // PORTB is output, all pins
   PORTB = 0x00;   // LED's are off
   for (;;) {
      PORTB ^= 0xFF;   // invert all the pins
      _delay_ms(500); // wait some time
   }
   return 0;
}

pi@raspberrypi ~/ATtiny85/blinky $ 
```

Program the ATtiny85:

```
cd ~
mkdir ATtiny85
cd ATtiny85
mkdir blinky
cd blinky
```

create the blinky.c file and add the following code

```
nano blinky.c
///////////////////////
#define F_CPU 1000000L
#include <avr/io.h>
#include <util/delay.h>

int main(void)
{
DDRB = 0xFF; // PORTB is output, all pins
PORTB = 0x00; // Make pins low to start

for (;;) {
PORTB ^= 0xFF; // invert all the pins
_delay_ms(100); // wait some time
}
return 0;
}
///////////////////////
```

add this code to a Makefile file

```
nano Makefile
//////////////////////
MCU=attiny85
```

```
AVRDUDEMCU=t85
CC=/usr/bin/avr-gcc
CFLAGS=-g -Os -Wall -mcall-prologues -mmcu=$(MCU)
OBJ2HEX=/usr/bin/avr-objcopy
AVRDUDE=/usr/local/bin/avrdude
TARGET=blinky
all :
$(CC) $(CFLAGS) $(TARGET).c -o $(TARGET)
$(OBJ2HEX) -R .eeprom -O ihex $(TARGET) $(TARGET).hex
rm -f $(TARGET)

install : all
sudo gpio -g mode 22 out
sudo gpio -g write 22 0
sudo $(AVRDUDE) -p $(AVRDUDEMCU) -P /dev/spidev0.0 -c linuxspi -b 10000
-U flash:w:$(TARGET).hex
sudo gpio -g write 22 1
noreset : all
sudo $(AVRDUDE) -p $(AVRDUDEMCU) -P /dev/spidev0.0 -c linuxspi -b 10000
-U flash:w:$(TARGET).hex

fuse :
sudo gpio -g mode 22 out
sudo gpio -g write 22 0
sudo $(AVRDUDE) -p $(AVRDUDEMCU) -P /dev/spidev0.0 -c linuxspi -b 10000
-U lfuse:w:0x62:m -U hfuse:w:0xdf:m -U efuse:w:0xff:m
sudo gpio -g write 22 1

clean :
rm -f *.hex *.obj *.o
///////////////////////
```

(Sorry, the instructables text editor has destroyed all the tabs and spacing in the code above, the Makefile will not work without tabs in the correct place)

To compile type:
make
To compile and upload code to the ATtiny:
make install
To optionally send fuses:
make fuse

This program oscillates ALL 5 pins as outputs so the LED should flash.
We are programming the ATtiny directly using avr-libc.
To change the setup of the ATtiny get fuses from :
https://github.com/kcuzner/avrdude (https://github.com/kcuzner/avrdude)

Comments

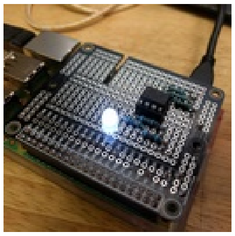
We have a be nice comment policy.
Please be positive and constructive.

I Made it! Add Images Post Comment

ahernandez5 (/member/ahernandez5/) made it!

2017-05-30 Reply

This was so much fun, I even added a serial port pin to receive data, and blinky!

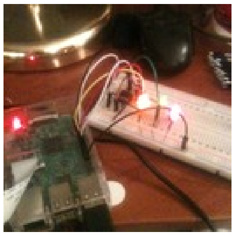


(https://cdn.instructables.com/FTF/VX9F/J3AFDZOL/FTFVX9FJ3AFDZOL.LARGE.jpg)

keaaw (/member/keaaw/) made it!

2017-03-10 Reply

Thanks a lot for this! I really love the ATtiny series, smaller is better! And with a Rasbperry Pi system, I'm using a tiny linux computer to program a tiny chip! No Arduino or ISPs necessary.

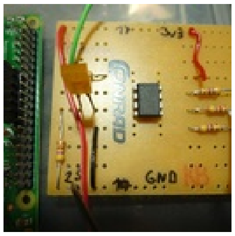


(https://cdn.instructables.com/FON/7L2V/J02QP8NV/FON7L2VJ02QP8NV.LARGE.jpg)

molle17031976 (/member/molle17031976/) made it!

2017-01-09 Reply

For me this tutorial is essential and the best take-off ramp to the ATtiny-issue. Thank you very much.



(https://cdn.instructables.com/FO1/XPHE/IXQFRPDP/FO1XPHEIXQFRPDP.LARGE.jpg)

HenricD (/member/HenricD/) made it!

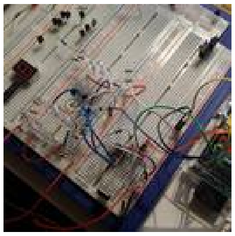
2016-12-19 Reply

Thanks!

Managed to get started coding on my ATtiny right away!

With inspiration I made a simple christmas decoration lamp which twinkles really nice :)

https://github.com/henricdav/blinky (https://github.com/henricdav/blinky)



(https://cdn.instructables.com/F4M/XTD8/IWWFHEBB/F4MXTD8IWWFHEBB.LARGE.jpg)

prb3333 (/member/prb3333/) ▸ HenricD (/member/HenricD/) 2016-12-19 Reply

Thank you, and everyone, for comments. It's great to know that people find it useful. Cheers!

STEMROX (/member/STEMROX/) 2016-05-11 Reply

I am lost when it comes to coding... Where should the TABS go? Can someone please post a picture of the correct code?

prb3333 (/member/prb3333/) ▸ STEMROX (/member/STEMROX/) Reply 2016-05-11

Hi,

I have the code on Github now:

https://github.com/paulbarber/ATtiny85_RaspberryPi... (https://github.com/paulbarber/ATtiny85_RaspberryPi/tree/master/blinky)

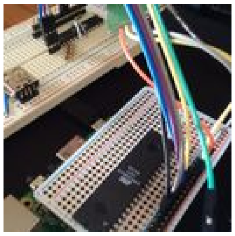
Cheers!

NeilC41 (/member/NeilC41/) 2016-03-13 Reply

Thankyou so much! I had purchased 10x ATtiny84a's and had no idea, until I stumbled upon this. Much Appreciated:)

斐郑 **(/member/%E6%96%90%E9%83%91/)** made it! 2016-02-07 Reply

Thank you so much, I have made it work with my Atmega8-16PI and will try with the Atmega8535-16PI



(https://cdn.instructables.com/F9P/1LZ2/IKCV3V26/F9P1LZ2IKCV3V26.LARGE.jpg)

LarsO2 (/member/LarsO2/) 2016-01-16 Reply

Hi, this helped me to get going, but you should update/change one thing.

After reading this post: http://kevincuzner.com/2013/05/27/raspberry-pi-as-... (http://kevincuzner.com/2013/05/27/raspberry-pi-as-an-avr-programmer/)I noticed that the 'linuxspi' programmer by default uses the GPIO pin 25 for reset, so your reset wrapper script is not needed. Else, great work ;)

breagan22 (/member/breagan22/) 2015-12-24 Reply

Outstanding!!

PavelH2 (/member/PavelH2/) 2015-10-01 Reply

Thank you.

grimw (/member/grimw/) 2015-08-30 Reply

This is a fantastic tutorial, and it really helped me get started. I'd like to point out a couple things I learned along the way, however. I don't know about at the time you were doing this project, but the "linuxgpio" configuration for avrdude exists in /etc/avrdude.conf on the RPi (after you install avrdude, of course); you simply uncomment the lines and set the pins to what they should be. Second, because the AVR and the RPi both run at 3.3v, I didn't find any need to use resistors for anything other than the LED itself, and that was only to protect the LED. And finally, if you hook the LED's positive pole to one of the SPI pins, the LED will happily blink for you as the RPi uploads data, which is cool and gives some nice feedback!

grimw (/member/grimw/) ▸ grimw (/member/grimw/) 2015-08-30 Reply

Oh, and I used this with an ATmega88; I'm sure you can use it with any ATmega that supports 3.3v operation, which, for all I know, is all of them.

iugamarian (/member/iugamarian/) 2015-08-13 Reply

Works with atmega8. When I poweroff the Raspberry Pi the Reset pin GPIO 22 gets to zero, stopping the program on the microcontroller. I added a pullup resistor of 10 k to 3.3 v to GPIO22 and now it works after Pi poweroff too :) . Thank you. I made it but I don't yet have a camera / photo capable phone so no images...

haneja (/member/haneja/) 2015-06-20 Reply

Is it possible to program Atmega8-8PU using this?

jollyviking (/member/jollyviking/) 2015-05-05 Reply

Thank you! Thank you! Thank you! My L.E.D.s are flashing, and my stepper motor is spinning. I could not have gotten this far this fast without your VERY helpful Instructable!

AlejandroL2 (/member/AlejandroL2/) 2015-04-09 Reply

i dont know nothing about makefile, could you help me, which is the path where the file should be, i have to paste or create a new file ?

alvaro.vaz.18 (/member/alvaro.vaz.18/)

hi im having this error"28:*** target pattern contains no '%' .Stop

2015-03-16 Reply

Can anyone help me with this....

bacoms (/member/bacoms/) 2015-03-12 Reply

On the strength of getting blinky to work following this instructable (thanks for that), I thought I'd use what I'd learnt to install Atmel's TWI Bootloader on a ATTiny85 (Atmels Application Note: AVR112).

Unfortunately I fell at the first hurdle running the 'make' with error

'fatal error: ioavr.h: No such file or directory'

Has anyone done this successfully? If so, some pointers on getting passed this would be much appreciated.

bacoms (/member/bacoms/) ▸ bacoms (/member/bacoms/) 2015-03-12 Reply

Problem solved by replacing lines:

#include <ioavr.h>
#include <inavr.h>

with:

#include <avr/io.h>

Running the 'make' again gives error message:

Common_Define.h:164:1: error: unknown type name ‘__no_init’

Line 164 in Common_Define.h is:

__no_init uint8_t pageBuffer[PAGE_SIZE];

Can anyone throw some light on why code from Atmel doesn't seem to work?

david.manciet (/member/david.manciet/) 2015-03-03 Reply

I try to upload this code instead of blinky but i got a lot of error and i don't know why if someone can help me

https://www.instructables.com/files/orig/F95/O60L/I...
(https://www.instructables.com/files/orig/F95/O60L/I45HOS1E/F95O60LI45HOS1E.txt)
from https://www.instructables.com/id/MASON-JAR-MOOD-LAM...
(https://www.instructables.com/id/MASON-JAR-MOOD-LAMP/?ALLSTEPS)

PS : It works perfectly when i upload blinky

prb3333 (/member/prb3333/) ▸ david.manciet (/member/david.manciet/) Reply 2015-03-08

Hi,

That code uses the Arduino code libraries ("digitalWrite" etc.). You can program these devices that way but I do not cover it here.

I am using the avr-libc library. By the way this gives you the possibility to have three pwm outputs from the ATtiny85. I have mood light code I have used in the past, it's here:

https://gist.github.com/paulbarber/fd0467d912f3a2d...
(https://gist.github.com/paulbarber/fd0467d912f3a2d644bd)

I hope you can work out from the comments what the circuit should be like.

Cheers,

Paul.

david.manciet (/member/david.manciet/) ▸ prb3333 (/member/prb3333/) 2015-03-09 Reply

Thanks for the help i understand why it doesn't work now.
I gonna try to directly upload the hex generate by arduino and if it doesn't work i gonna use your code to make mine.

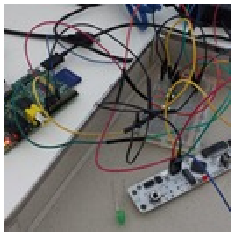
Thanks again

spirail (/member/spirail/) made it! 2014-07-02 Reply

Works good! thankyou!

I made it work with my custom ATmega32u4 board. I had to supply it with an external 3.3v supply.

Later I plan to supply with wiht USB power (5V) but im still waiting for my level converters to come in the mail.



(https://cdn.instructables.com/F25/G4WH/HX42Q8AI/F25G4WHHX42Q8AI.LARGE.jpg)

vpopov1 (/member/vpopov1/) 2014-06-23 Reply

Hello and Thank you for your Tutorial.

But i have a few problems.

When i run make install it flashes ok, but i receive failure:

"avrdude verification error content mismatch"

and "avrdude safe mode lfuse changed"

What i doing wrong ?

I am trying to programm attiny45

vpopov1 (/member/vpopov1/) 2014-06-23 Reply

Hello and thank you for tutorial.

But i have a problem. I would like to program Attiny45 and when i run make install ist says :

avrdude verification error content mismatch

avrdude safe mode louse changed, was 62 and now is 0

and when i set y to change fuse back then it freeze ...

Could you please help me

mva111 (/member/mva111/) made it! 2014-06-07 Reply

Very straightforward tutorial. Used it with the ATtiny25 and worked well. Thank you!

(https://cdn.instructables.com/F6V/RNME/HW3K8YTQ/F6VRNMEHW3K8YTQ.LARGE.jpg)

fubar1 (/member/fubar1/) 2014-05-12 Reply

Many thanks! Your suggestion was spot on!!!!

fubar1 (/member/fubar1/) 2014-05-11 Reply

Hello, I've followed your tutorial and get as far as 'Makefile'. When I run the 'make' command I get .....

'Makefile:9: *** missing separator. Stop'

Any help would be appreciated.

prb3333 (/member/prb3333/) ▸ fubar1 (/member/fubar1/) 2014-05-11 Reply

Probably a lack of tabs. You need a hard tab, or a ';' before each command. Where there is a gap at the beginning of the line in the picture of the code, make sure this is a hard tab. See also:

http://stackoverflow.com/questions/14109724/makefi... (http://stackoverflow.com/questions/14109724/makefile-missing-separator)

Hope this helps.

prb3333 (/member/prb3333/) ▸ fubar1 (/member/fubar1/) 2014-05-11 Reply

Probably a lack of tabs. You need a hard tab, or a ';' before each command. Where there is a gap at the beginning of the line in the picture of the code, make sure this is a hard tab. See also:

http://stackoverflow.com/questions/14109724/makefi... (http://stackoverflow.com/questions/14109724/makefile-missing-separator)

Hope this helps.

ajohnson134 (/member/ajohnson134/) made it! 2014-05-03 Reply

Great tutorial! All of the steps were very clear. The wiring diagrams were a big help. Took me no time to set this up. Everything worked on my first try (well... almost everything. I accidentally mixed up two ports).

(https://cdn.instructables.com/FNZ/IK9V/HUQATMQ5/FNZIK9VHUQATMQ5.LARGE.jpg)

ajohnson134 (/member/ajohnson134/) ▸ ajohnson134 (/member/ajohnson134/) 2014-05-03 Reply

And technically I'm using ATtiny25, but switching processors was actually fairly trivial. :)

harnon (/member/harnon/) 2014-01-24 Reply

I wasn't fully done before...I just got it all working now. Great !
It is flashing.

Great job man.

one more 'enter' for separate "cd ~ mkdir ATtiny85" and "cd blinky" for easy line copying newbies like me ;o)

You did the most understable tuto I got for this. Hope you will have success.

prb3333 (/member/prb3333/) ▸ harnon (/member/harnon/) 2014-01-25 Reply

Thanks, clarified that more. As with all instructables:

The more we use it, the more we improve it.

Cheers,

harnon (/member/harnon/) 2014-01-24 Reply

Hello,

Really nice step by step instructable. I searched hours for it.Thanks !

2 suggestions for perfect cut & paste tutorial :

"sudo raspi-config" instead of " sudo raspi-coinfig"

"cd ~ && git clone git://git.drogon.net/wiringPi" instead of "cd ~ git clone git://git.drogon.net/wiringPi"

This allowed me to continue the tuto.

Have a nice day.

prb3333 (/member/prb3333/) ▸ harnon (/member/harnon/) 2014-01-24 Reply

Hi,

Thanks for the comments, glad it was of help. Corrections made.

Cheers,

dan3008 (/member/dan3008/) 2014-01-22 Reply is it ironic that I want to do this, so I can make an USB AVR programer? (https://www.instructables.com/id/VUSBTiny-AVR-SPI-Programmer/?&sort=ACTIVE&limit=40&offset=40#DISCUSS) lol

prb3333 (/member/prb3333/) ▸ dan3008 (/member/dan3008/) 2014-01-22 Reply

Yes, I for one did not think it was possible to interface to the ATtiny via USB to make an AVR programmer. That's awesome!

dan3008 (/member/dan3008/) ▸ prb3333 (/member/prb3333/) 2014-01-22 Reply

Yeh it is, and alot cheaper than a standard usb AVR programmer. Its limited on what chips it works with, but should save alot of time and effort in the future :D

More Comments