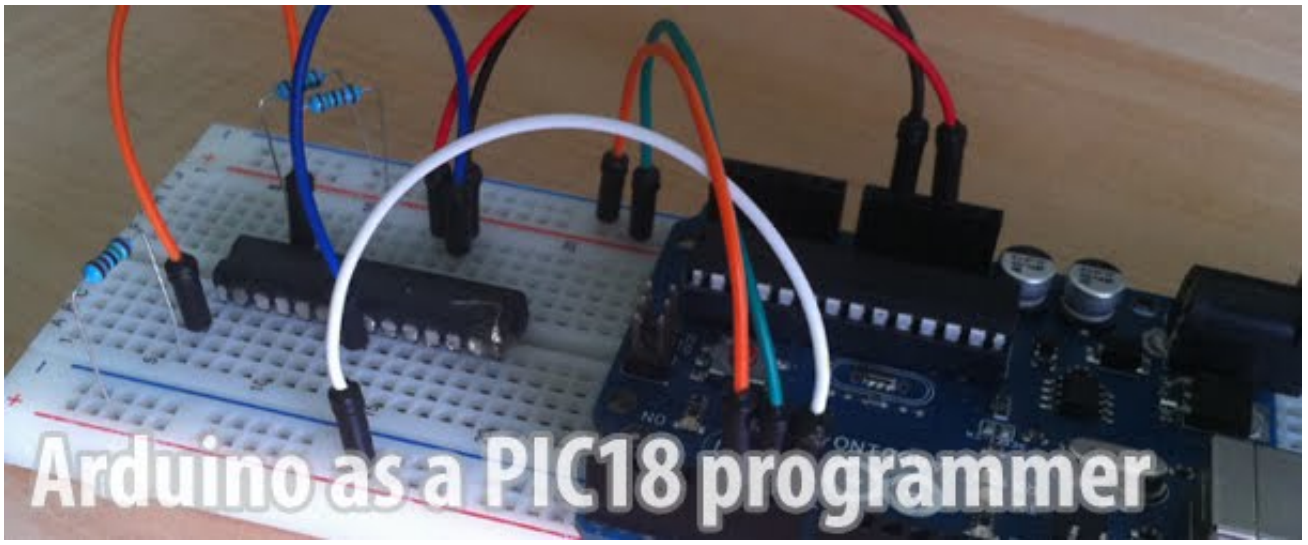
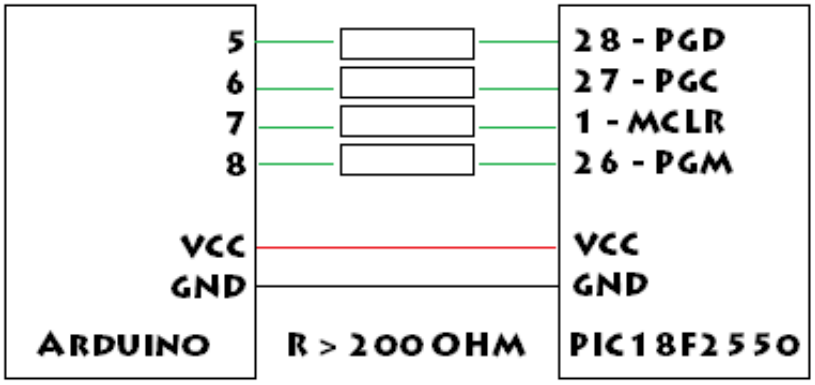


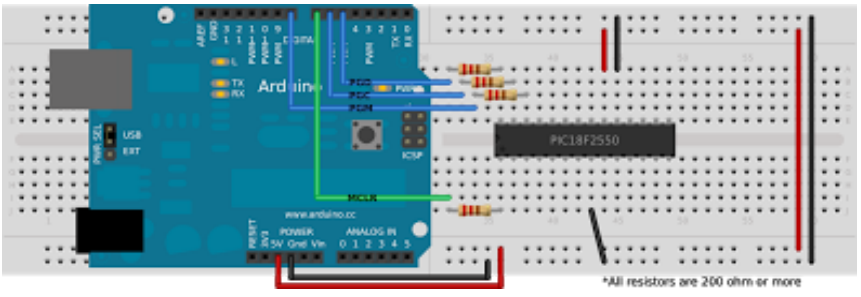
So I get my hand on a PIC18F2550.Unfortunately I don't have a PIC programmer, and I'm reluctant to build a parallel/printer port programmer, so I made one using my Arduino.

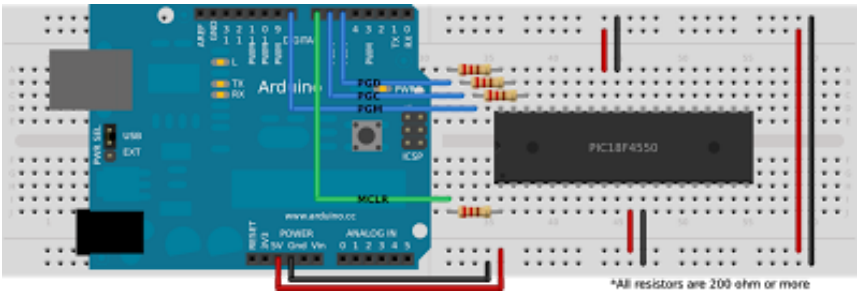
The PIC18F2/4 family support low-voltage programmer which makes easier to program via the Arduino.Moreover the programming specifications are well written and quite easy to understand.I wrote my sketch to work with PIC18F2550 only, but with a minor changes it can be work for the whole family as well (a full list of the family can be found at the end).

Connecting the arduino to the PIC is quite simple:



simplified diagrams





thanks to ***Jose Carlos Granja*** *for these Arduino friendly diagrams*

note! the pins should be connected through a resistor, I used 1.2K. anything over 200R will be just fine.

UPDATE #5

version 0.3 is out
fixes that arduino was not detected unless it was over COM3

Thanks to **Nickson Yap** that pointed me out the bug,
as well as helped me recompile the project

the newest version can be downloaded at the bottom of the page

UPDATE #4

The project has been ported by ***Harry Braviner*** to PIC18F2420, you can find well written tutorial over Harry's blog
http://pleasantlyclueless.blogspot.co.uk/2013/11/using-arduino-as-pic-programmer.html

This 'dummies guide' might be much understandable for beginners.
Unless you are planning on porting it to another microcontroller, I'd suggest you to take a look at his post.

UPDATE #3

The programmer was ported to Linux!
I've test it myself, if you manage to get it working **let me know in the comments**

The adoption for Linux was made by ***Jose Carlos Granja***
https://bitbucket.org/JoseFuzzNo/arduino-as-pic18f-programmer-for-linux

UPDATE #2

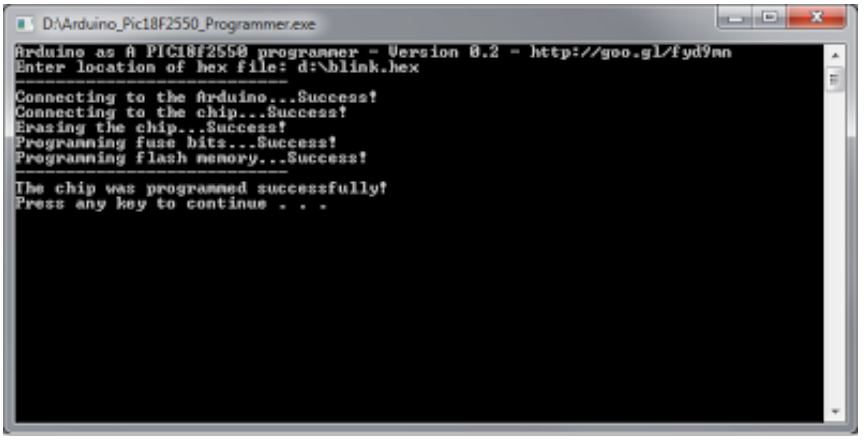
The source code was added to BitBucket, here is a link to the repo
https://bitbucket.org/kirill578/arduino-as-pic18f-programmer/src

UPDATE #1

Version 0.2 is out!
Now you can program your Pic just by selecting a .hex file
if you have any questions please leave your message in the comment section

instructions

1.Download the new version
2.upload the sketch to the Arduino
3.connect the pic18f2550 to the arduino
4.locate your hex file (you can use the blink example, just connect an led to one of the pins)



Memory

In order to program the chip, upload the sketch to your arduino(can be downloaded at the end)
but before we start let's see how the memory is organized:

0000h-07FFh	**Boot block**
0800h-1FFFh	**Block 0**
2000h-3FFFh	**Block 1**
4000h-5FFFh	**Block 2**
6000h-7FFFh	**Block 3**
8000h-1FFFFFh	*Unimplemented
200000h-200007h	*ID Location
200008h-2FFFFFh	Config Space - irrelevant
300000h-30000Dh	**config/fuse bits**
30000Eh-3FFFFD	Config Space - irrelevant
3FFFFEh	**Device ID1**
3FFFFFh	**Device ID2**

*Unimplemented = read as '0'
*ID Location - are not implemented in my code

Interfacing with the programmer

now, let see how to interface with the programmer.the command listed bellow will be send to the arduino via the serial console.

Read command: the following command read 32bytes works for 0x0 - 0x3FFFFF:

R< 6digit address >X
example:
R30000DX

Write command:the following command writes 32bytes of data, it work for 0x0-0x7FFF, the address that you are writing to have have to be even on the second digit and have to be 0 on the first digit,other wise it won't work:

W< 4digit address >< 32 bytes of data>X
example:
W7FE0FFFFFFFFFFFFFFFFFFFFFFFF0001F9EF3FF0926A936A946A89688A688B68FFD7X
the following command writes:
0x007FE0: FFFFFFFFFFFFFFFFFFFFFFFF0001F9EF3FF0926A936A946A89688A688B68FFD7

Erase command: preform a factory reset. **reconnect the power afterward!!!**

EX

Write config/fuse bits: sets one byte at a time:

C<Number of fuse, in HEX><one bytes>X
example:
CA2CX
the following writes:
0x30000A: 2C

FLASH AN LED

you can send the following command in order to make the device flash all it's PORT at 1Hz

EX (**reconnect the power afterward!!!)**
C108X
W0000F6EF3FF0FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFX
W7FE0FFFFFFFFFFFFFFFFFFFFFFFF0001F9EF3FF0926A936A946A89688A688B68FFD7X

the following commands erase the chip, set the chip run via the internal clock
and write the following

0x000000:F6EF3FF0FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
0x007FE0:FFFFFFFFFFFFFFFFFFFFFFFF0001F9EF3FF0926A936A946A89688A688B68FFD7

this was taken from the follwing hex file:

HEX file

```
:04000000F6EF3FF0E8
:107FEC000001F9EF3FF0926A936A946A89688A6893
:047FFC008B68FFD7B8
:00000001FF
```

which is the compiled version of this:

Source code

```
#include <p18f2550.h>

unsigned int i;

void main(){
  TRISA = 0;
  TRISB = 0;
  TRISC = 0;

  while(1){
  LATA = 255;
  LATB = 255;
  LATC = 255;
  for(i=0;i<16000;i++);
  LATA = 0;
  LATB = 0;
  LATC = 0;
  for(i=0;i<16000;i++);
  }

while(1){}

}
```

I know that is takes a lot of time to convert an HEX file in to commands that will write the compiled code, There is

needed somesort of GUI that will do it, I'm not planning to write it in the near further, If you do I'd be glad if you post in over here

that's it feel free to comment.

PIC18F2XXX/4XXX Family:

- PIC18F2221 • PIC18F2580 • PIC18F4480
- PIC18F2321 • PIC18F2585 • PIC18F4510
- PIC18F2410 • PIC18F2610 • PIC18F4515
- PIC18F2420 • PIC18F2620 • PIC18F4520
- PIC18F2423 • PIC18F2680 • PIC18F4523
- PIC18F2450 • PIC18F2682 • PIC18F4525
- PIC18F2455 • PIC18F2685 • PIC18F4550
- PIC18F2458 • PIC18F4221 • PIC18F4553
- PIC18F2480 • PIC18F4321 • PIC18F4580
- PIC18F2510 • PIC18F4410 • PIC18F4585
- PIC18F2515 • PIC18F4420 • PIC18F4610
- PIC18F2520 • PIC18F4423 • PIC18F4620
- PIC18F2523 • PIC18F4450 • PIC18F4680
- PIC18F2525 • PIC18F4455 • PIC18F4682
- PIC18F2550 • PIC18F4458 • PIC18F4685
- PIC18F2553