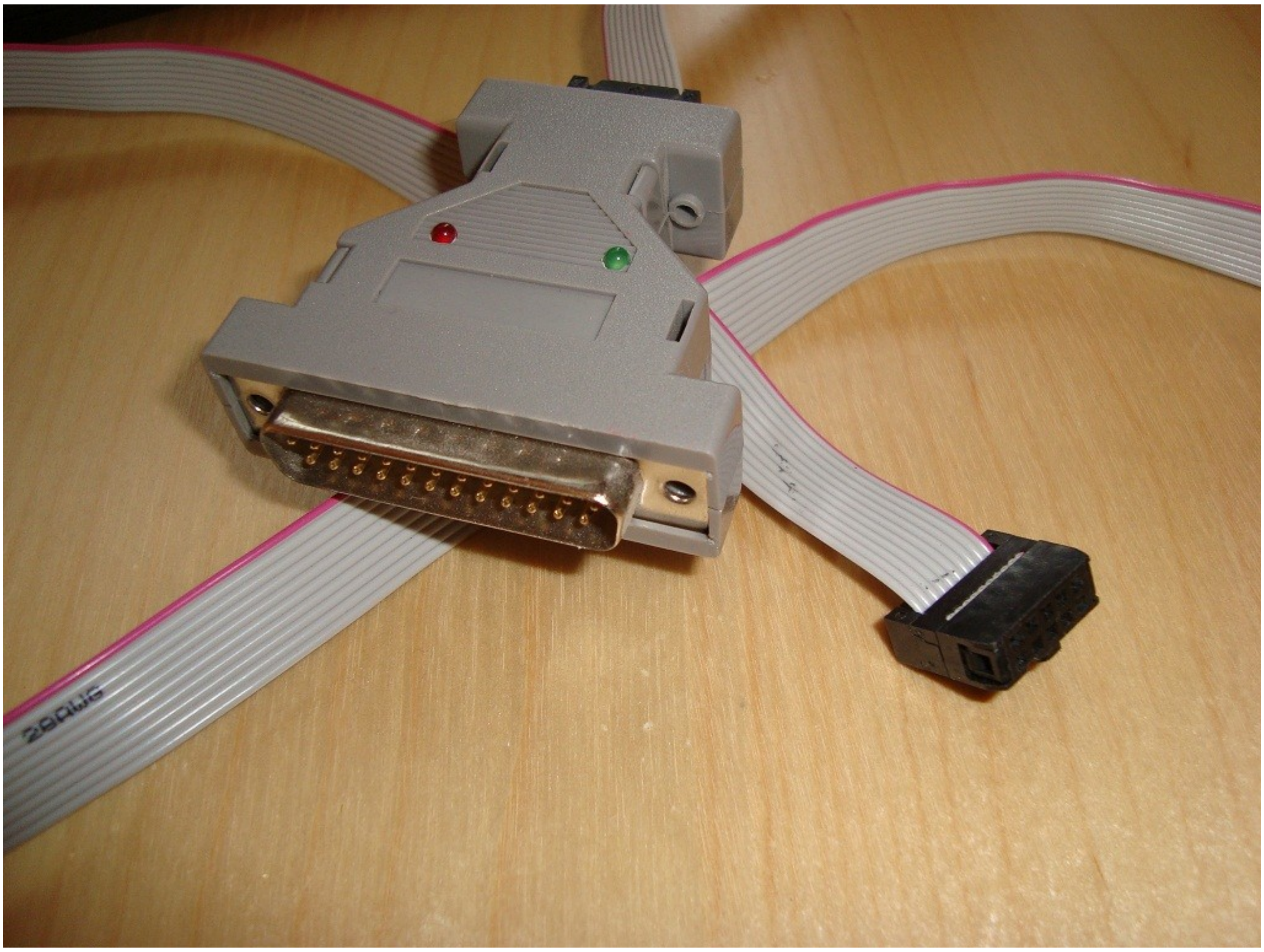
Gamecube or PSX analog stick in N64 controller

I. Preparations

In order to program the Atmega8 or Attiny24 microcontroller you'll need an ISP programmer for your PC's parallel port, just like this one:

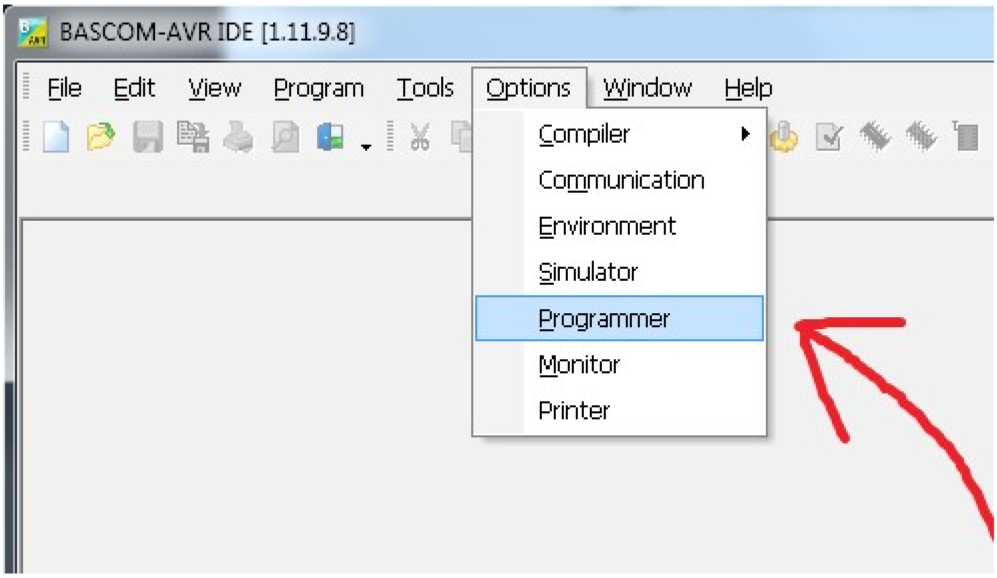


It's important that the programmer is STK200-compatible. These programmers can be found on eBay easily.

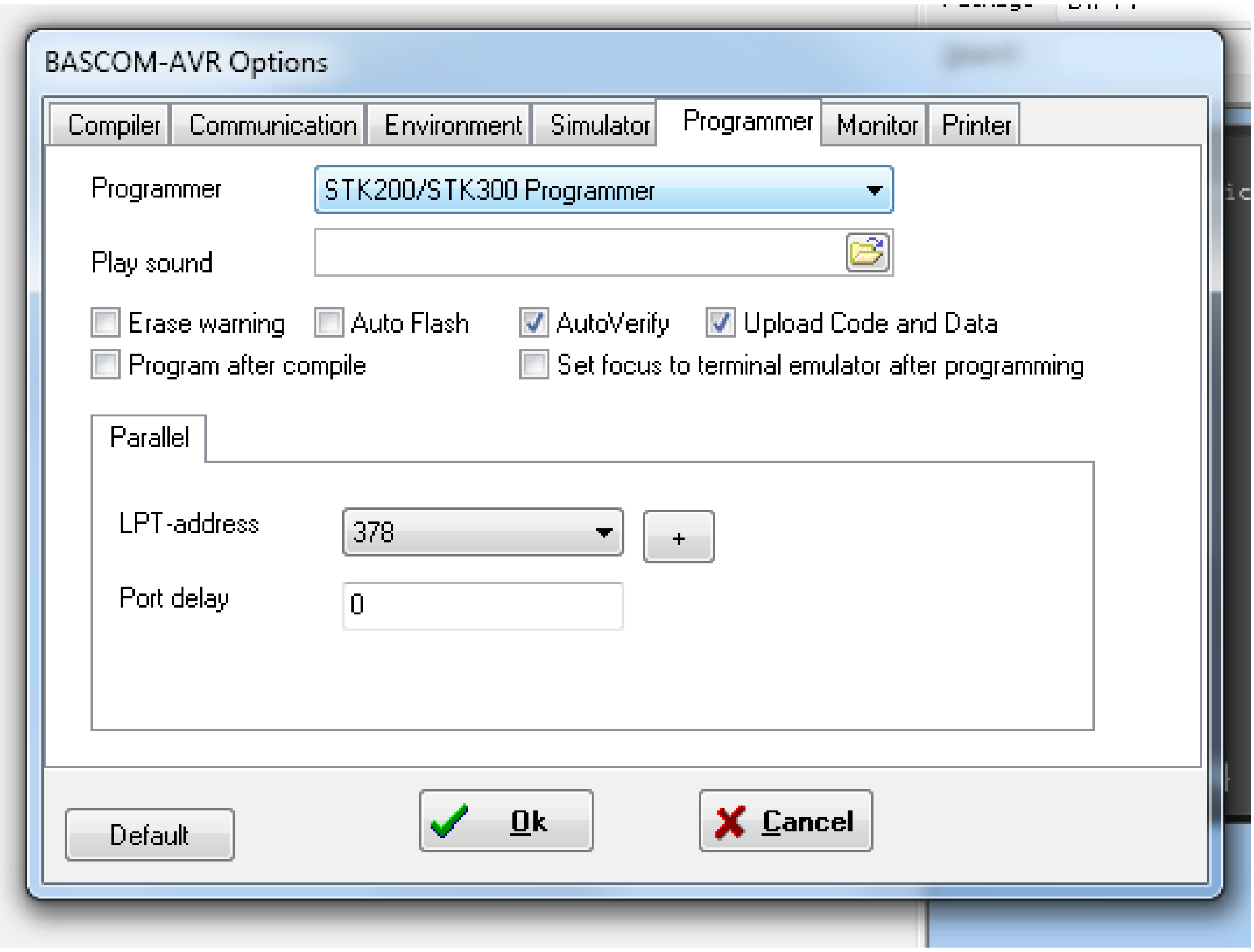
The software we'll be using is BASCOM-AVR by MCS. A fully-functional demo version (up to 4KB code) of BASCOM-AVR can be obtained at http://www.mcselec.com/index.php?option=com_content&task=view&id=14&Itemid=41

II. Setting up BASCOM

Start BASCOM and go to „Options“ → „Programmer“



Select „STK200/STK300 Programmer“ and confirm.



Programmer: STK200/STK300 Programmer

Play sound:

□ Erase warning □ Auto Flash ☑ AutoVerify ☑ Upload Code and Data

□ Program after compile □ Set focus to terminal emulator after programming

Parallel

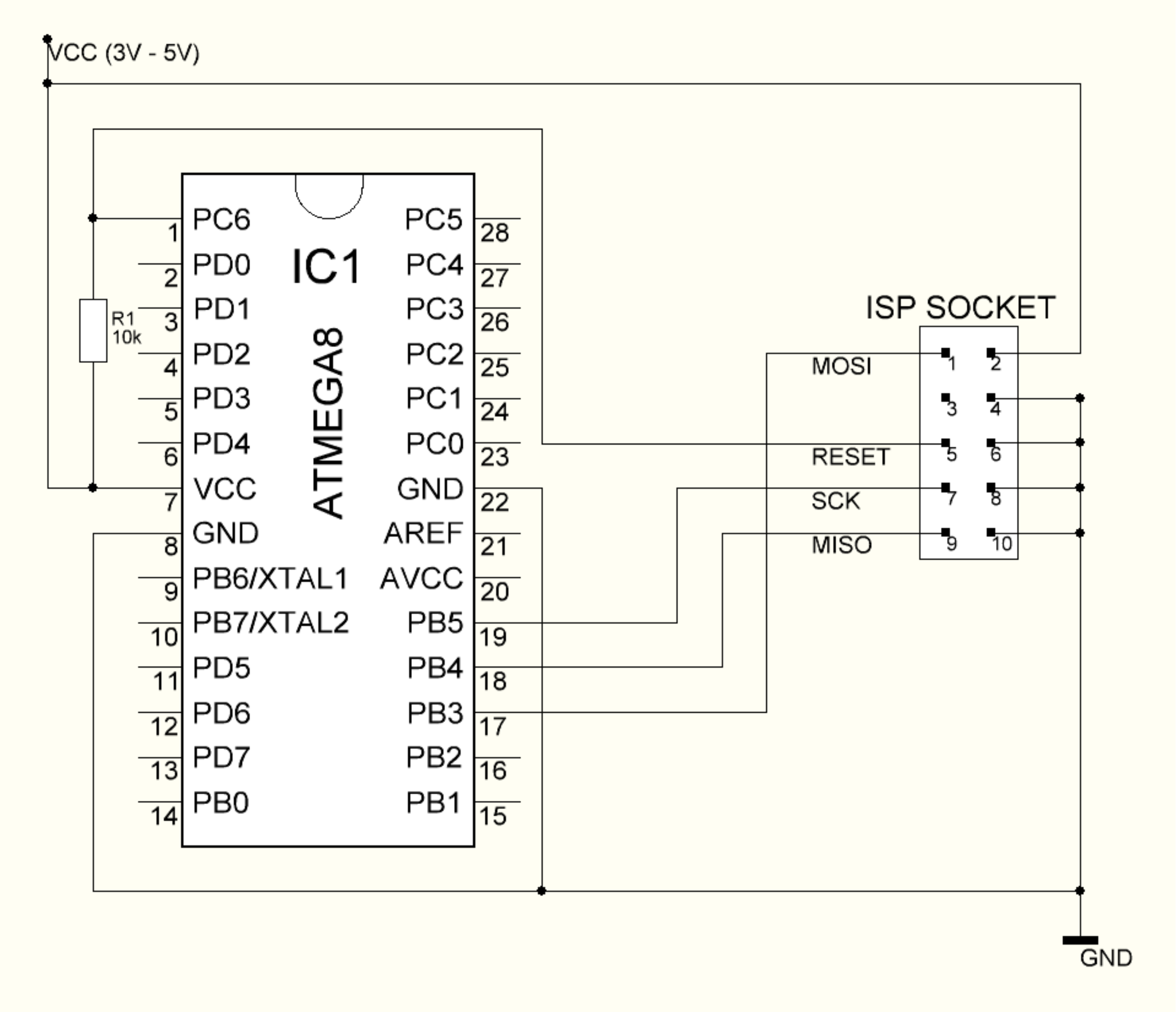
LPT-address: 378 +

Port delay: 0

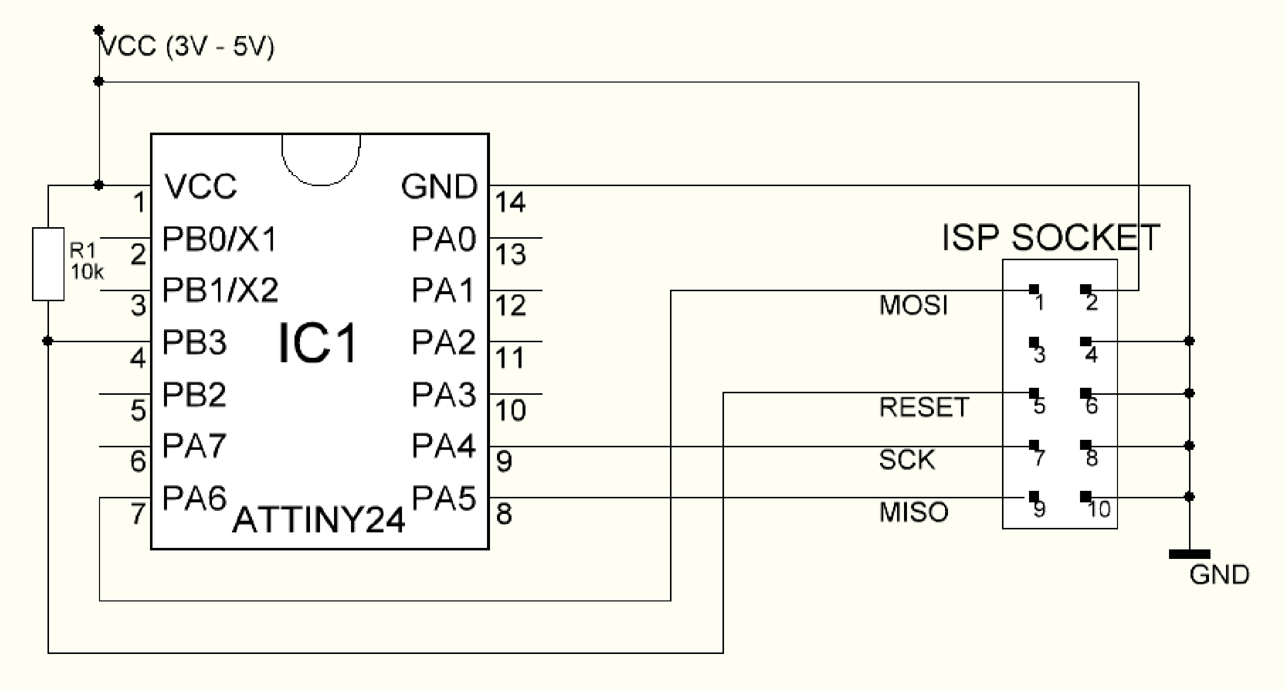
III. Schematics for programming the microcontroller

There are 2 different schematics depending on the microcontroller you are going to use:

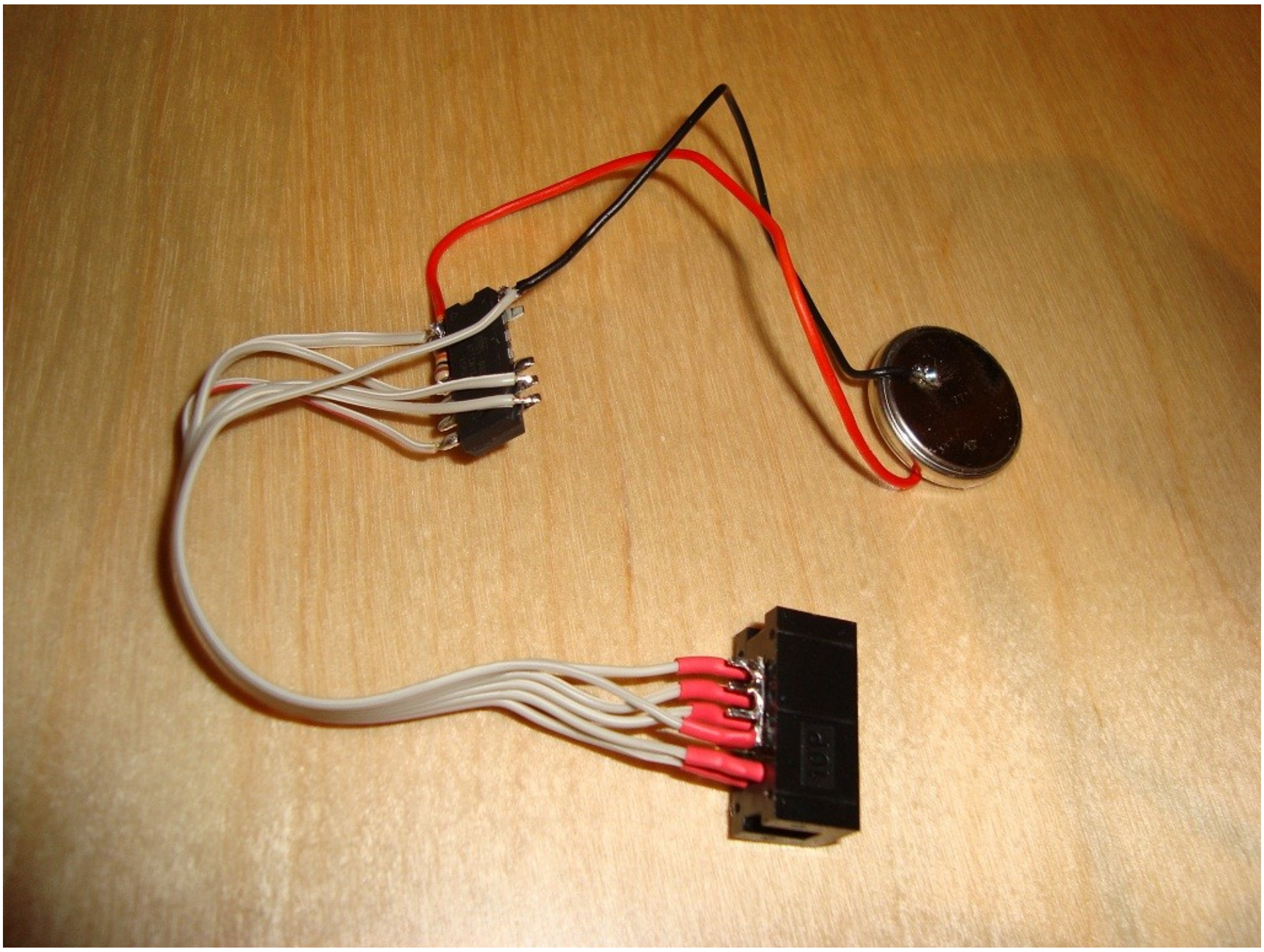
Atmega8:



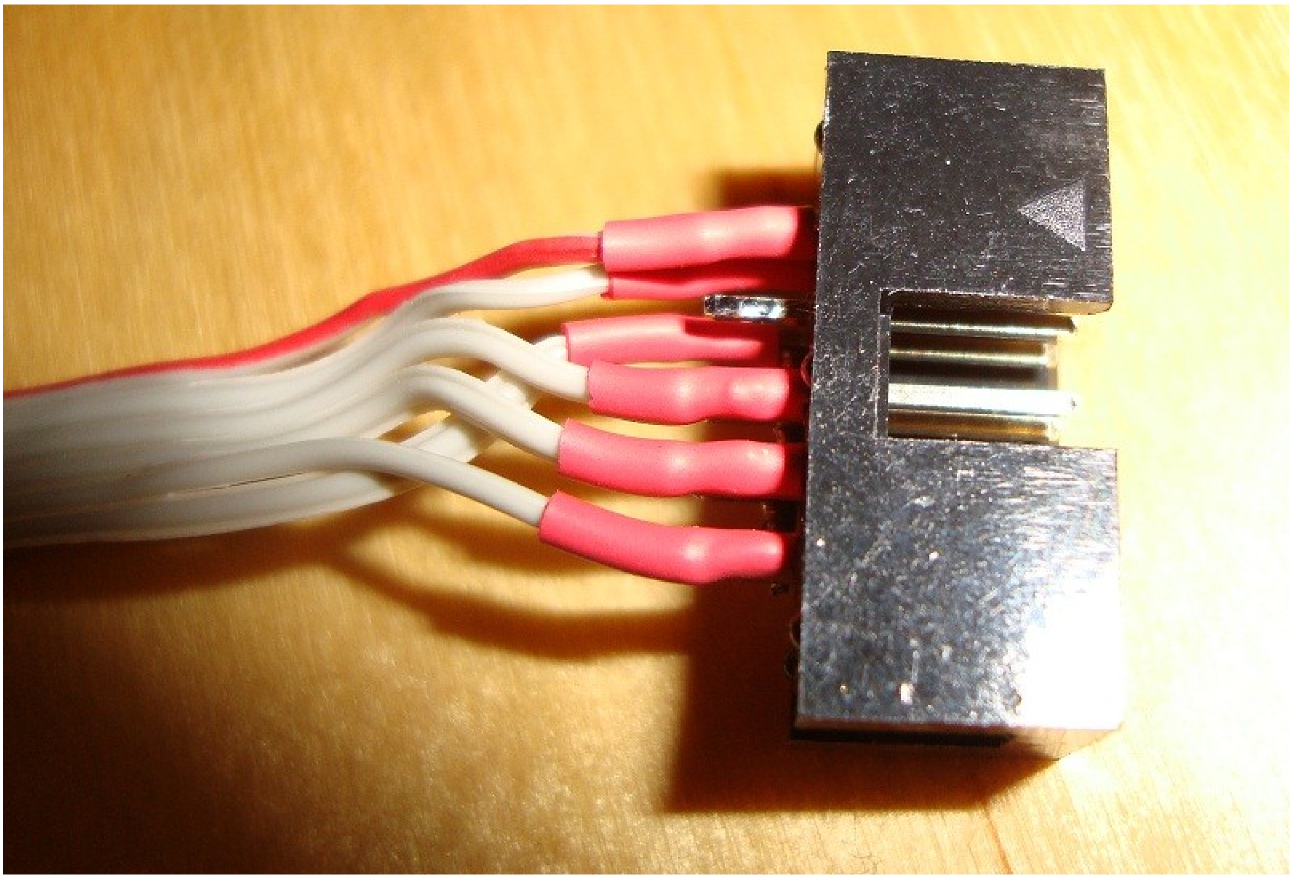
Attiny24:

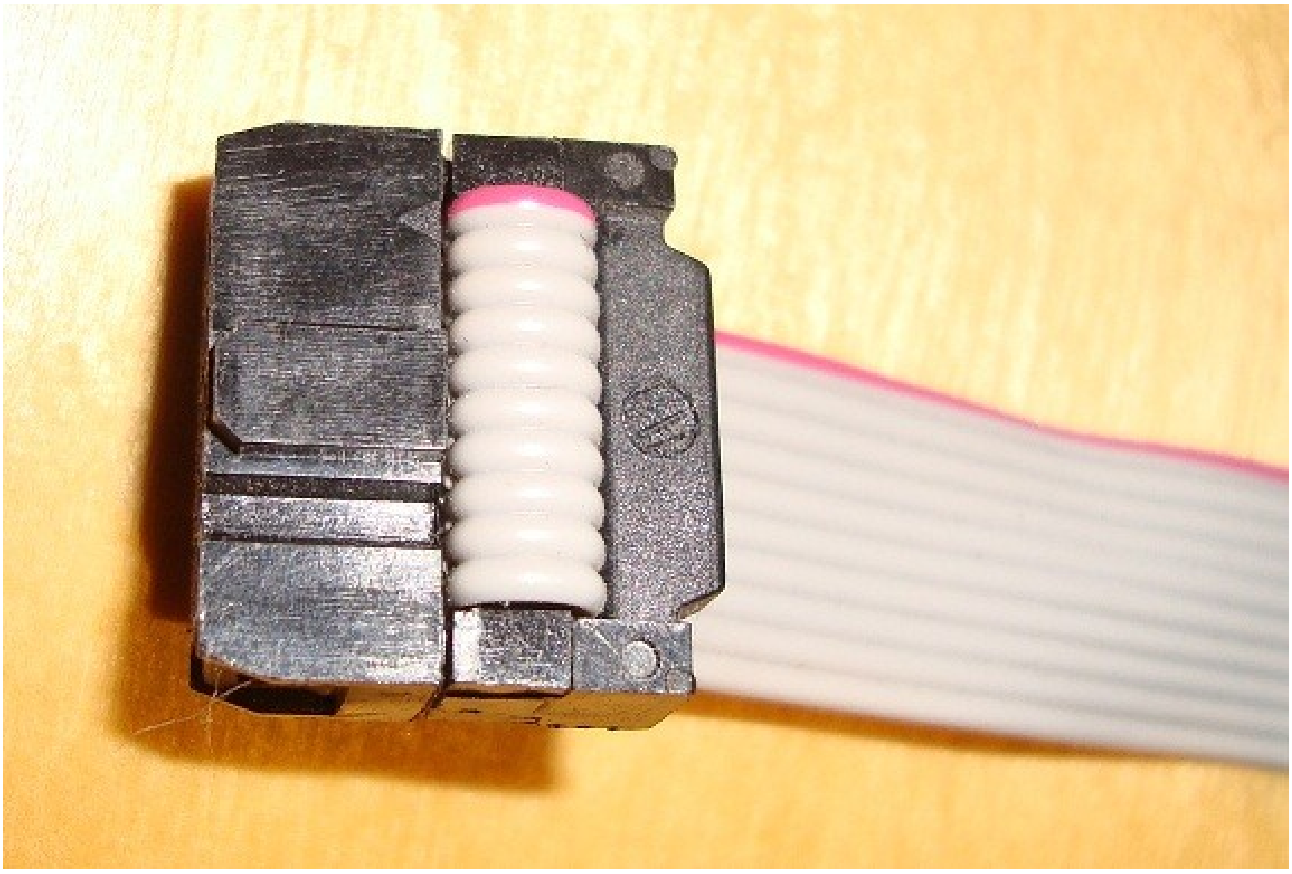


For programming the microcontroller you can also use a battery. Have a look how simple and compact the programming circuit for the Attiny24 can be:



Remember: The red wire of the programmer's ribbon cable is always pin 1. In addition on the ISP socket & ISP plug there should be a little arrow indicating pin 1.

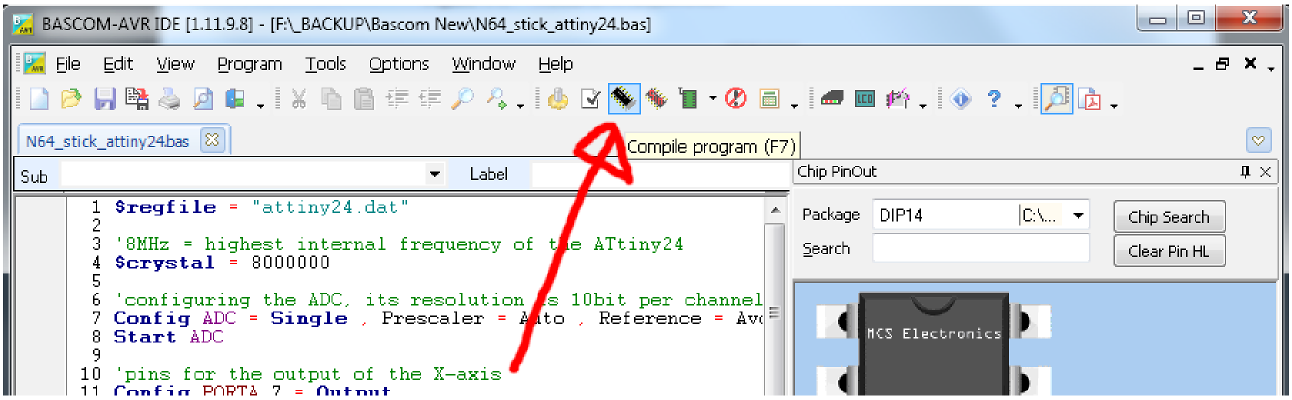




IV. Programming the microcontroller

Connect the programming cable to the parallel port of your PC and to the ISP socket of your microcontroller. Make sure you provide power to the microcontroller. Otherwise it can't be programmed. The programming cable does NOT provide power!

Start BASCOM and open the .bas source code file that fits your microcontroller.
Click on the „Compile“ symbol to generate a .hex file.

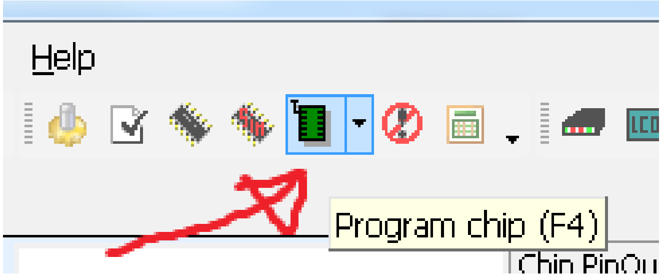


BASCOM-AVR IDE [1.11.9.8] - [F:_BACKUP\Bascom New\N64_stick_attiny24.bas]

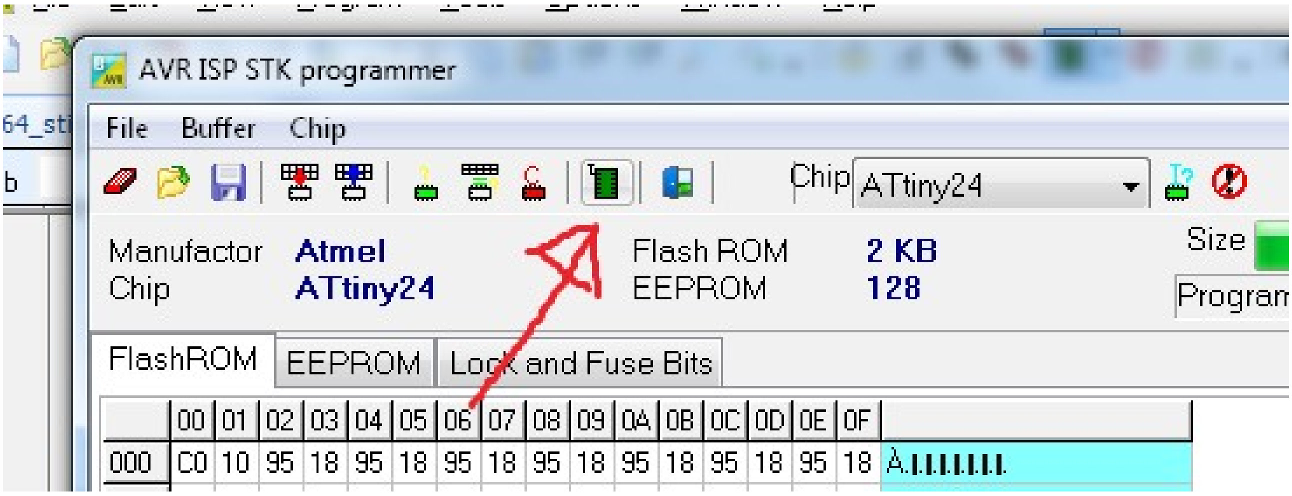
Package: DIP14 C:\...

```
$regfile = "attiny24.dat"

'8MHz = highest internal frequency of the ATtiny24
$crystal = 8000000

'configuring the ADC, its resolution is 10bit per channel
Config ADC = Single , Prescaler = Auto , Reference = Av
Start ADC

'pins for the output of the X-axis
Config PORTA.7 = Output
```

After that click on the green „Program chip“ symbol to bring up the programming window.



Here you'll find the same green symbol again. Push it and the microcontroller will be programmed.



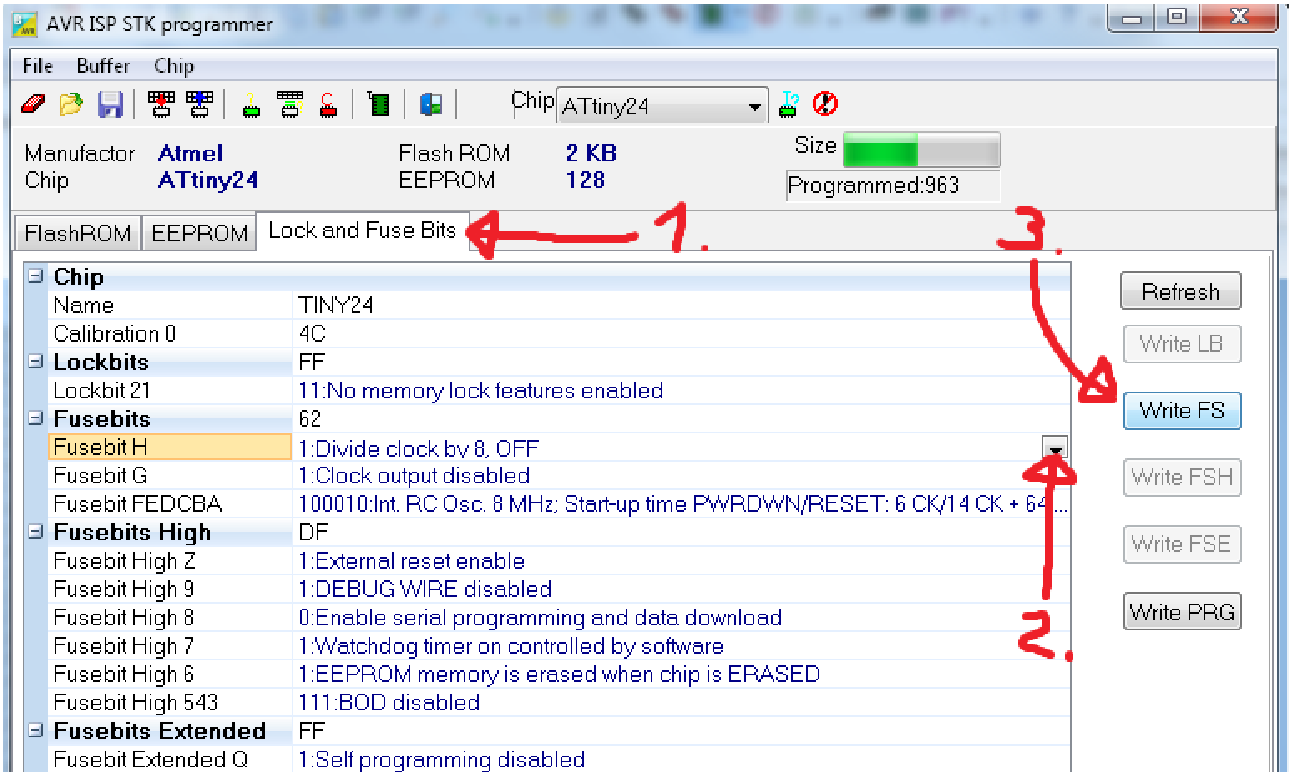
AVR ISP STK programmer

Chip: ATtiny24

Manufactor: Atmel
Chip: ATtiny24
Flash ROM: 2 KB
EEPROM: 128

```
    00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
000 C0 10 95 18 95 18 95 18 95 18 95 18 95 18 95 18  À.I.I.I.I.I.I.I.I.
```

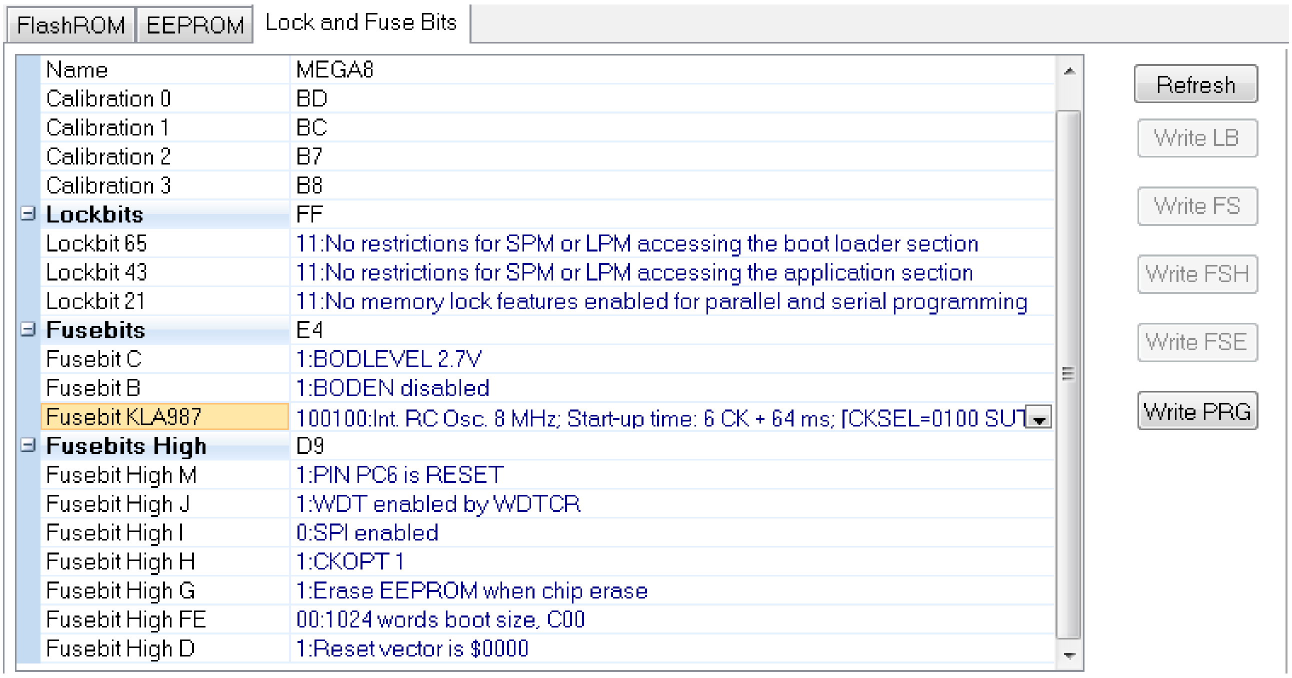
After the microcontroller has been successfully flashed there is one more thing to do in BASCOM. Click on the „Lock and Fuse Bits“ tab to see a list of programmed and unprogrammed fuse bits. For the Attiny24 you have to select „Divide clock by 8, OFF“ and push the „Write FS“ button.



Manufactor Atmel
Chip ATtiny24
Flash ROM 2 KB
EEPROM 128
Programmed:963

Chip	
Name	TINY24
Calibration 0	4C
Lockbits	FF
Lockbit 21	11:No memory lock features enabled
Fusebits	62
Fusebit H	1:Divide clock by 8, OFF
Fusebit G	1:Clock output disabled
Fusebit FEDCBA	100010:Int. RC Osc. 8 MHz; Start-up time PWRDWN/RESET: 6 CK/14 CK + 64...
Fusebits High	DF
Fusebit High Z	1:External reset enable
Fusebit High 9	1:DEBUG WIRE disabled
Fusebit High 8	0:Enable serial programming and data download
Fusebit High 7	1:Watchdog timer on controlled by software
Fusebit High 6	1:EEPROM memory is erased when chip is ERASED
Fusebit High 543	111:BOD disabled
Fusebits Extended	FF
Fusebit Extended Q	1:Self programming disabled

When using the Atmega8 your fuse bits should look like this:



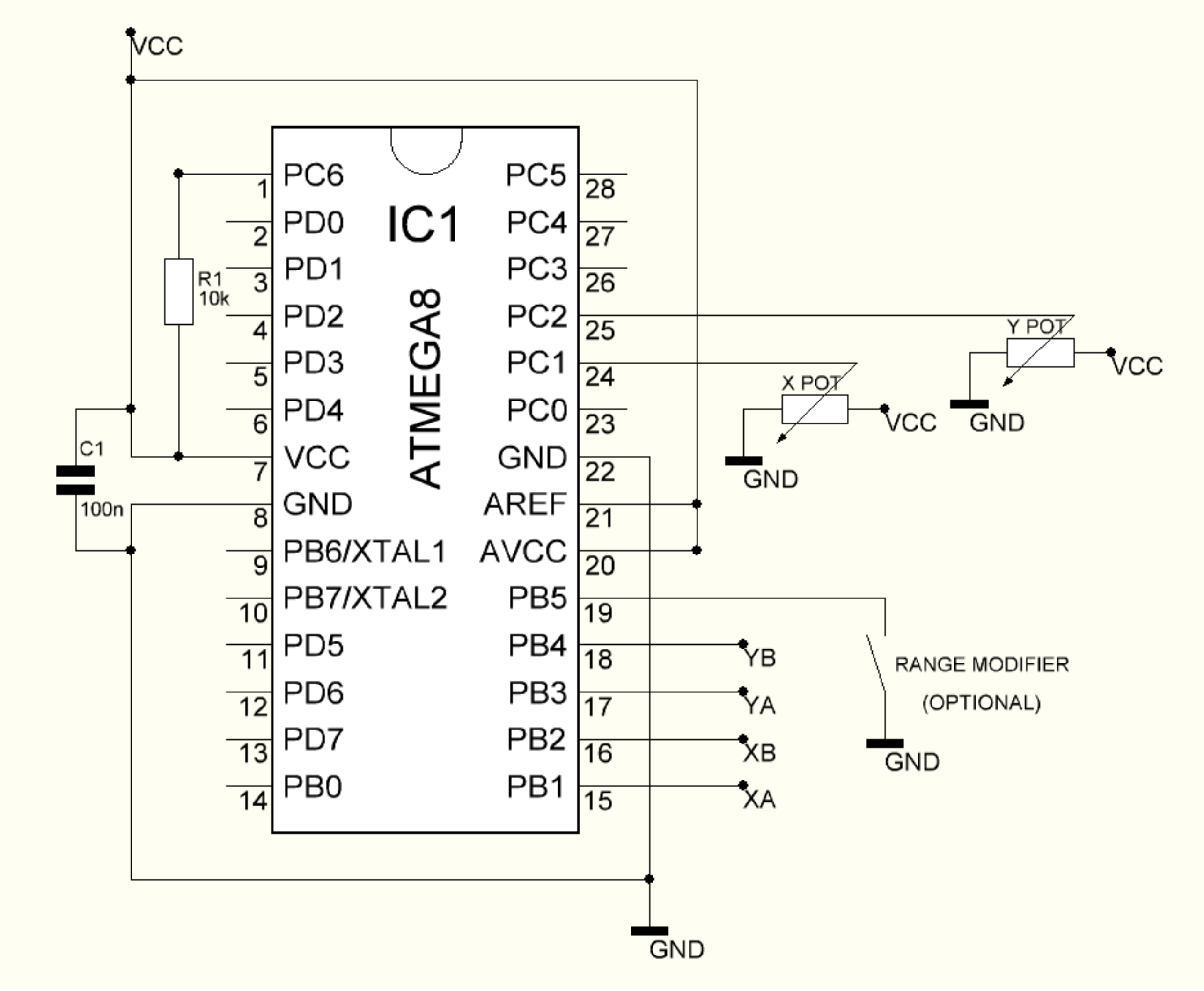
Name	MEGA8
Calibration 0	BD
Calibration 1	BC
Calibration 2	B7
Calibration 3	B8
Lockbits	FF
Lockbit 65	11:No restrictions for SPM or LPM accessing the boot loader section
Lockbit 43	11:No restrictions for SPM or LPM accessing the application section
Lockbit 21	11:No memory lock features enabled for parallel and serial programming
Fusebits	E4
Fusebit C	1:BODLEVEL 2.7V
Fusebit B	1:BODEN disabled
Fusebit KLA987	100100:Int. RC Osc. 8 MHz; Start-up time: 6 CK + 64 ms; [CKSEL=0100 SUT
Fusebits High	D9
Fusebit High M	1:PIN PC6 is RESET
Fusebit High J	1:WDT enabled by WDTCR
Fusebit High I	0:SPI enabled
Fusebit High H	1:CKOPT 1
Fusebit High G	1:Erase EEPROM when chip erase
Fusebit High FE	00:1024 words boot size, C00
Fusebit High D	1:Reset vector is $0000

After that you may close BASCOM and unplug the programming cable.

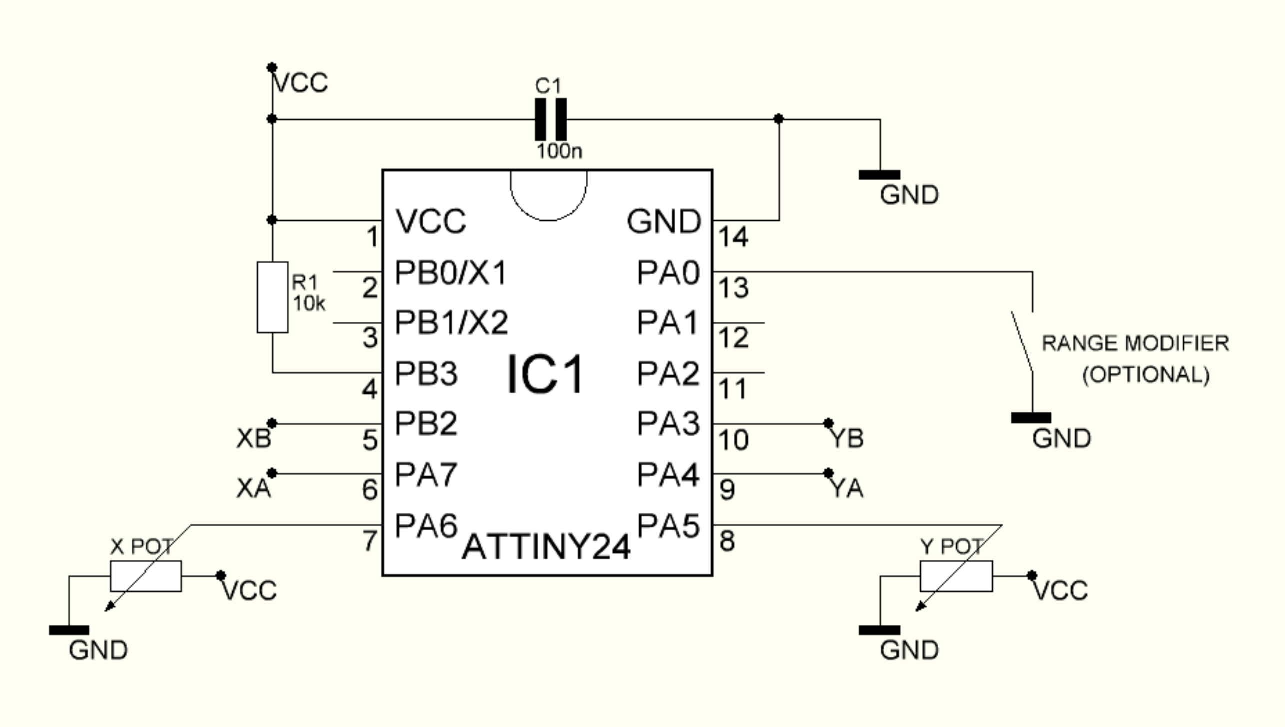
V. Installing the microcontroller into your N64 controller

I assume you've already attached your Gamecube or PSX analog stick to the stick unit of the N64 controller. So now you can finally install the microcontroller.
Again, the schematics for the Atmega8 and Attiny24 differ.

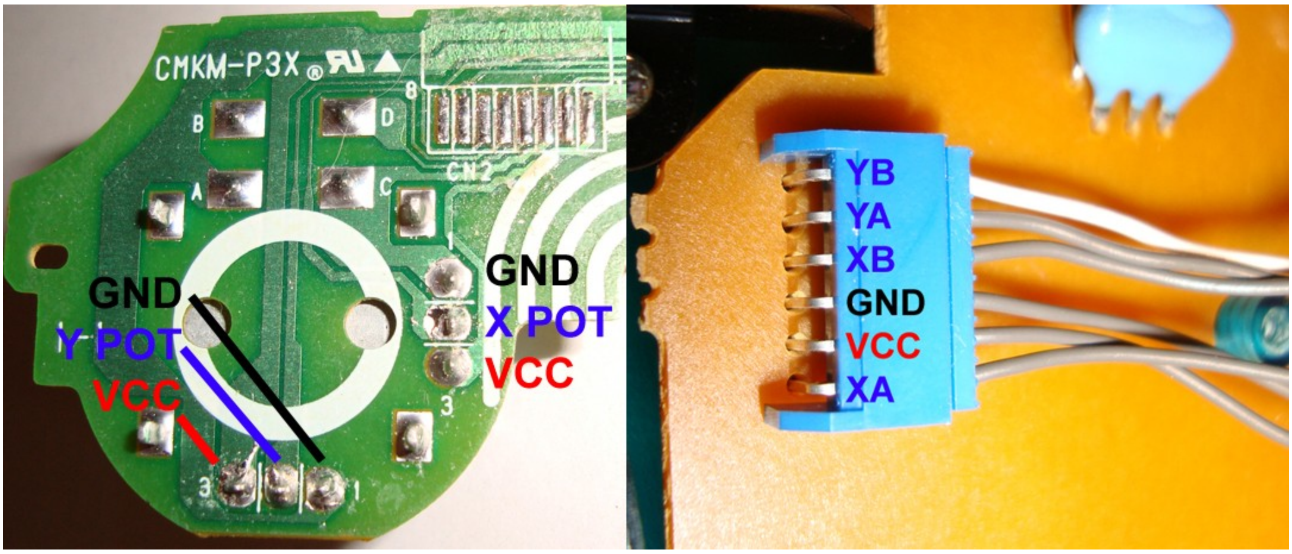
Atmega8:



Attiny24:



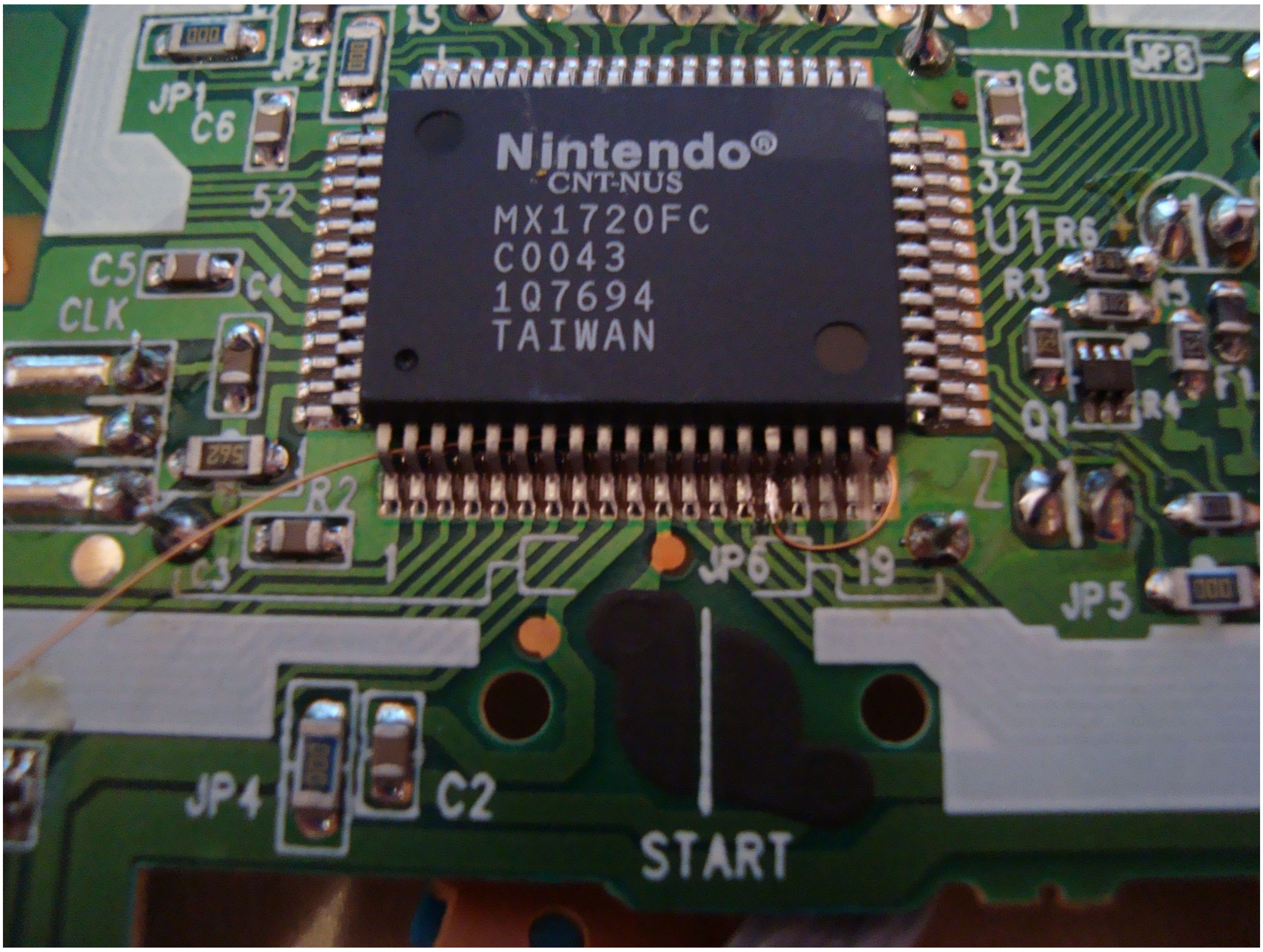
X Pot and Y Pot are the 2 potentiometers of your analog stick;
XA, XB, YA, YB, GND & VCC can be obtained from a connector on the N64 controller PCB:



You might have noticed the „range modifier“ switch in the schematics. This is absolutely optional. If Pin 19 on the Atmega8 or Pin 13 is driven low during power on, the range of the X- & Y-axis will be slightly higher than usual.

The default range are 160 steps / axis, the increased range will be 168 / axis.
160 steps should be enough but if you need more steps for whatever reason you now know what to do.

You can also use the B button of the N64 controller as range modifier switch. Simply attach a wire to Pin 15 of the CNT-NUS IC.



In that case you'd only need to hold down „B“ while turning on your N64 console to get the extended range of 168 steps / axis.

VI. Troubleshooting

Problem	**Solution**
X- and Y-axis are swapped!	Simply swap the „X Pot“ & „Y Pot“ wires.
Help, up is down and down is up / left is right & right is left!	Swap the wires for YA & YB or XA & XB depending on which axis is inverted.

Version 1.2 (corrected another error in the Attiny24 programming schematic)

Use at own risk.