

AVR MICROCONTROLLER ISP-PROGRAMMING

AVR IN SYSTEM PROGRAMMING

S'KHUMBUZO NDLOVU

06/2008

LAST EDITED : 10/2008

AVR ISP Programming requires only the SPI interface of the microcontroller as outlined below. That means all AVR processor with SPI port/ interface can be ISP programmed. In this article ATTINY26 is the target processor as an example. Most AVR microcontrollers can be programmed using this method. This is a very easy and economical way of programming microcontrollers. Only a parallel port programming cable is required (self made).

1. ISP-AVR PROGRAMMING.

AVR ISP Programming requires only the SPI interface of the microcontroller as outlined below. That means all AVR processor with SPI port/ interface can be ISP programmed. In this article ATTINY26 is the target processor as an example. Most AVR microcontrollers can be programmed using this method. This is a very easy and economical way of programming microcontrollers. Only a parallel port programming cable is required (self made). Only two application software's needs to be installed:

- gavrasm for assembling and compiling a hex file
- avrdude for downloading hex file to the target microcontroller

Compiling and downloading hex file to the microcontroller is done on the Linux terminal. All this

Figure 1. CORRECT AND VERIFIED CABLE CONNECTION.

was done using a computer running on Linux operating system. This programming method is applicable to all AVR microcontrollers, which means that the tiny26 used in this application can be substituted with other AVR processors.

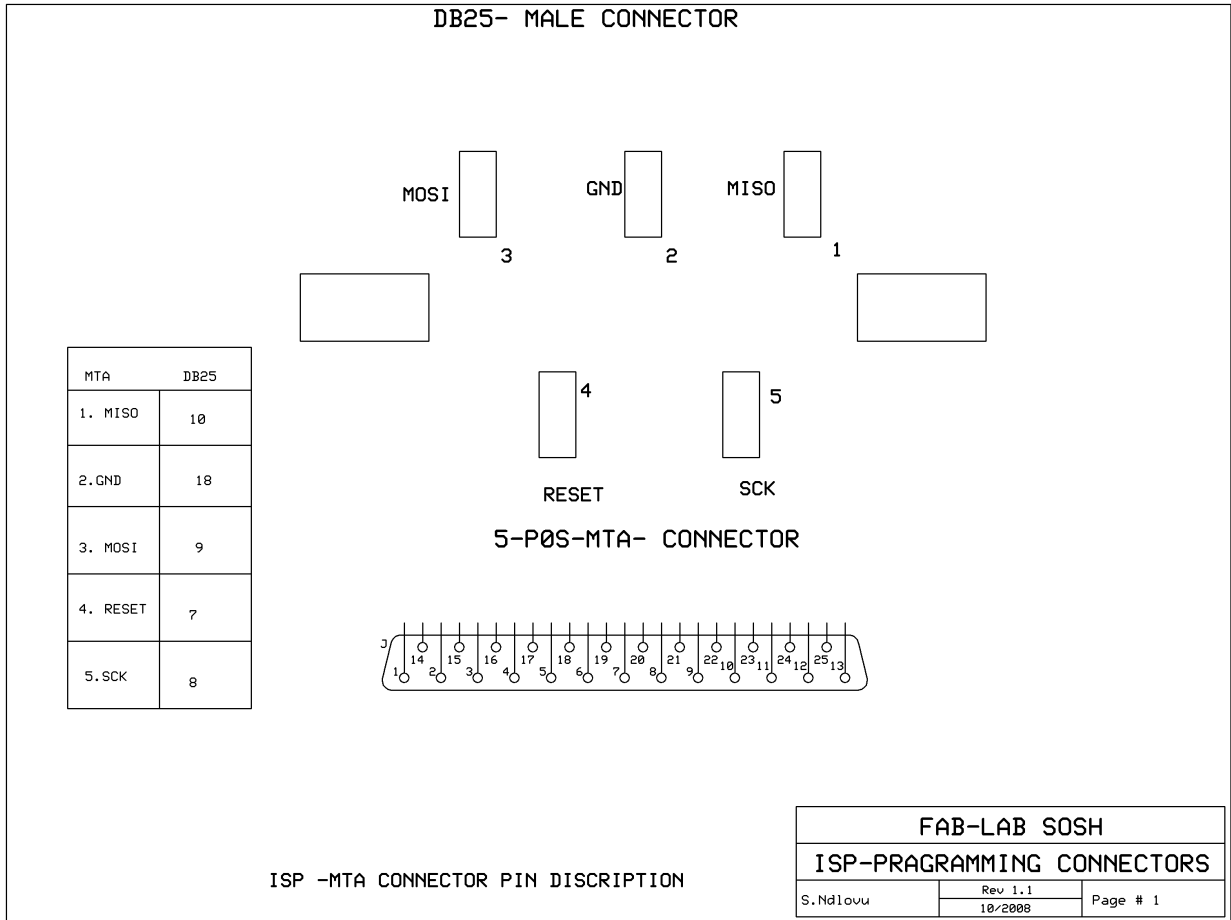


Figure 1:

1.1 ISP- PROGRAMING CABLE.

Only five lines (signals) are required to ISP-program a target microcontroller. ISP-programming uses- SPI interface to download the hex file into the target microcontroller. The target pins on the microcontroller are:-

- MOSI
- MISO
- SCK
- RESET
- GND

To make the cable you need:-

- 5-pos vertical AMP MTA connector (RIBON CABLE CONNECTOR)

- 5-pos vertical SMT pin header
- Ribbon cable
- DB25-male connector

How to make the cable:-

- Hold the 5-pos SMT MTA (male) pin header with its two pins facing to you, just like it is shown in the above diagram.
- Take a ribbon cable, connect it to the DB25-male connector/plug (only five wires are needed as showed in the drawing below)
- Only five pins of the DB25-male connector to the AMP MTA plug need to be wired together
- Start by connecting the required five wires to the DB25- male connector (pin 7, pin 8, pin 9, pin 10 & pin 18). I prefer soldering them directly to the pins(avoid overheating the connector since that can damage/deform the connector)
- Now connect the five wires from the DB25-male connector to the AMP MTA plug as outlined below
- Here are the pins of the two connectors to be wired together

DB25-pin 10 >> to AMP MTA plug pin 1

DB25-pin 18 >>to AMP MTA plug pin 2

DB25-pin 9 >> to AMP MTA plug pin 3

DB25-pin 7 >> to AMP MTA plug pin 4

DB25- pin 8 >>to AMP MTA plug pin 5

1.2 Programming the target AVR

1. First make the PCB and solder the components (good soldering means a faulty free circuit board, and no troubleshooting).
2. Power the circuit with a steady +5V power supply.
3. Plug the 5-pos AMP MTA plug to the target header (5-pos vertical MTA SMT pin header which is soldered on the target circuit board).

4. Plug the parallel DB25-male connector to the parallel port of the computer.
5. Now assemble the source code

5.1 Assembling the source code is done on the Linux terminal. It is better to have your source code file on the Desktop (if you are traditional Microsoft Windows User like me).

The following applications must be installed into your Linux computer before programming can be performed (for programming).

- `gavasm` (freeware, Linux version)
- `avrdude` (freeware, Linux version)

5.2 assembling and programming

- `gavasm name of the file.asm`
e.g. `gavasm toy car.asm`
- `avrdude -p t26 -c bsd -U flash:w:name of the file.hex` (remember to power the circuit)
e.g. `avrdude -p t26 -c bsd -U flash:w:toy car.hex`

notes: after assembling the source code a hex file will be created which is loaded to the target AVR processor during programming. `t26` is used when tiny26 is the target processor. Put AVR processor of your choice like tiny13- t13, tiny45- t45 etc.

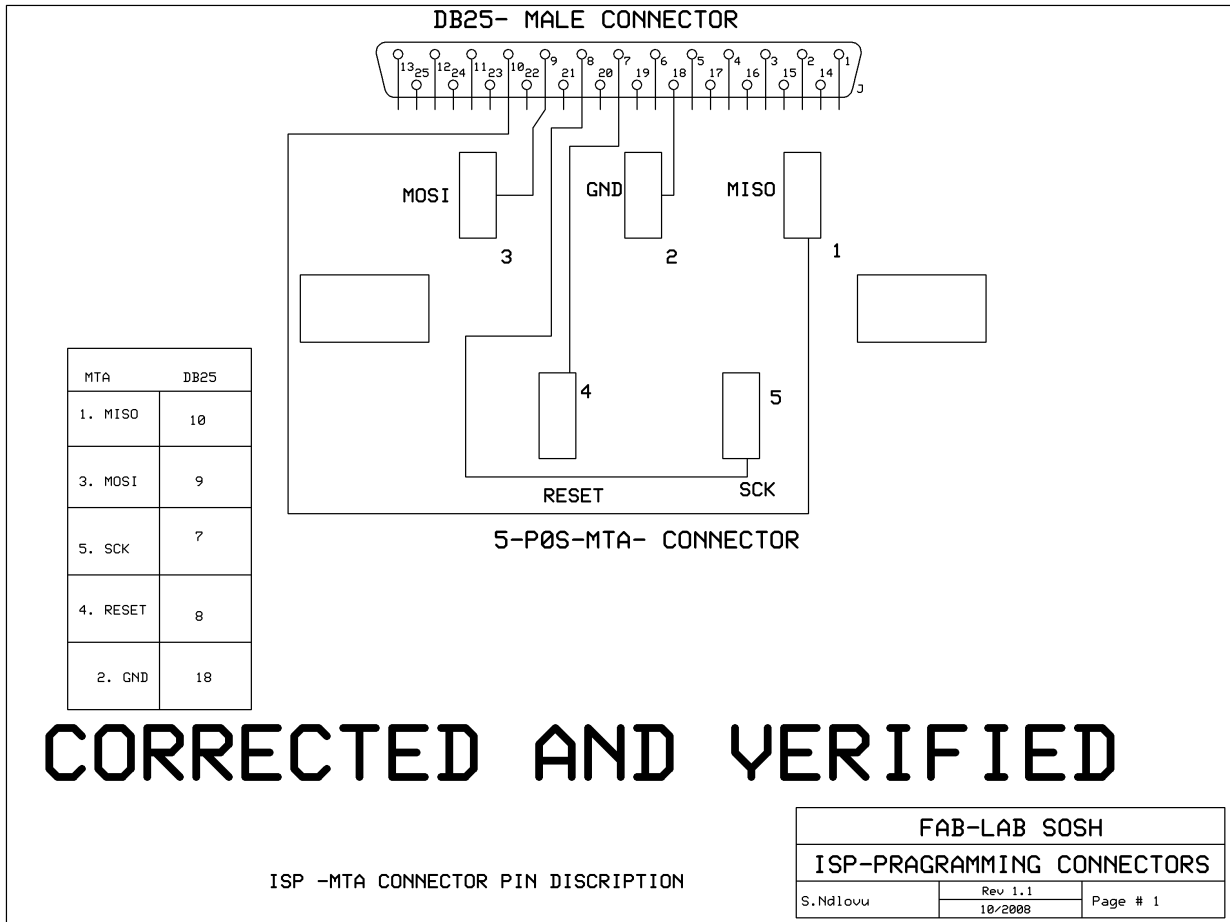


Figure 2:

3. ATTINY 26- AVR – ISP PROGRAMMING.

The diagram below shows how the 5-position MTA- connector/header should be connected to the target AVR (the ATTINY26).

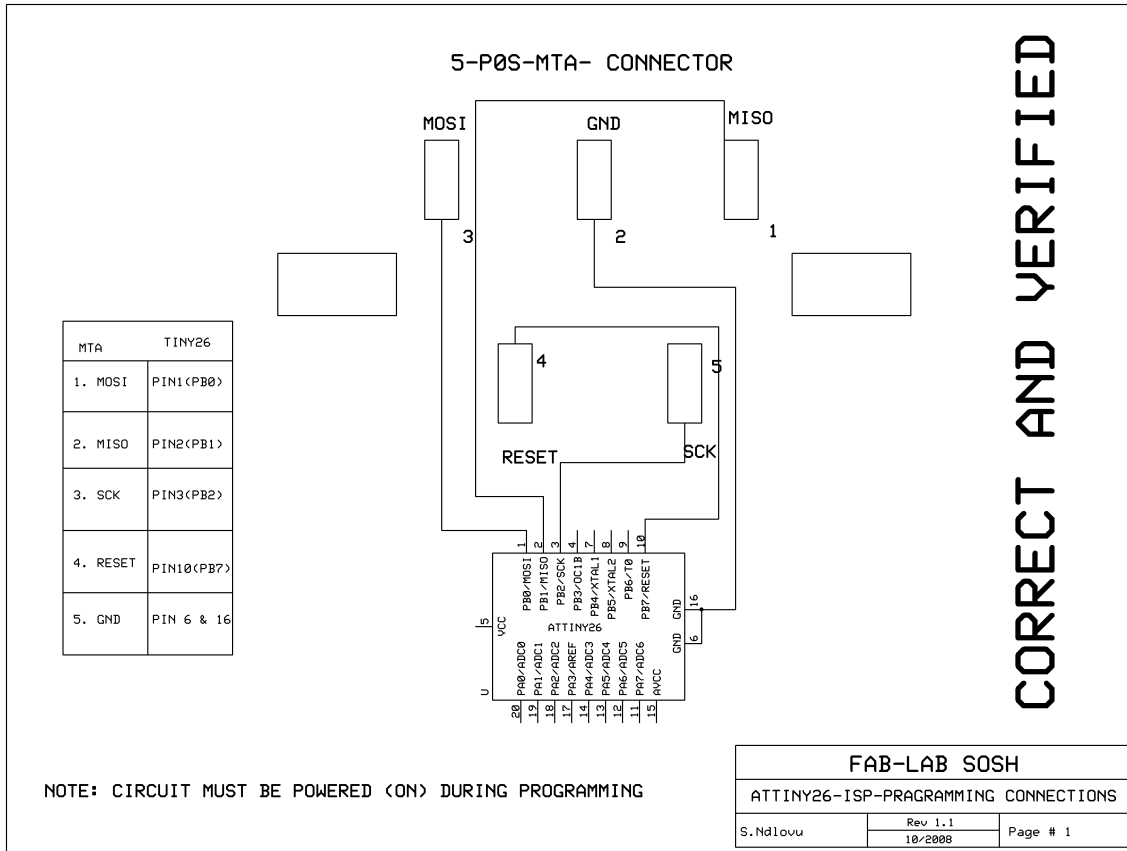


Figure 3: