2DOF firmware v1.0 for PWM AMC v1.5 board

Welcome to the manual of firmware version 1.0 of AVR Motion Controller v1.5 board.

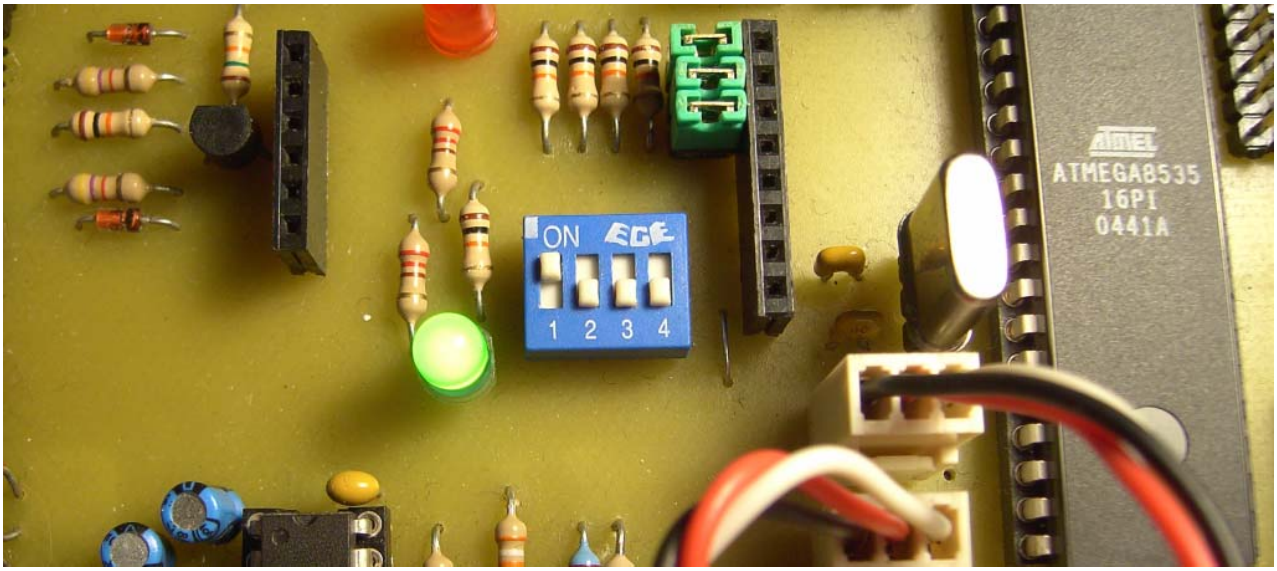
In this manual you will learn the basic functions of this firmware as well how to setup it to your motion simulator.

This firmware can be used with some specific crystals that each has its advantages and disadvantages. For example the 24 MHz crystal will produce a high quality 23 KHz PWM signal for driving the motors but will not always work with your PCB because of the huge overclocking that is involved (>50%!!!!). In the other hand using a 14.31818 MHz crystal will result in poor 13 KHz PWM quality signal for driving the motors but it will work in any case, even if the PCB is of no good quality, cause no overclocking in involved at all! A nice middle solution is to use a 22.1184 MHz or an 18.432 MHz crystal that are more stable and still have good quality PWM signal. But in all cases the servo control performance will remain the same!

Here are the filenames of the firmware and the crystals you should use them with:

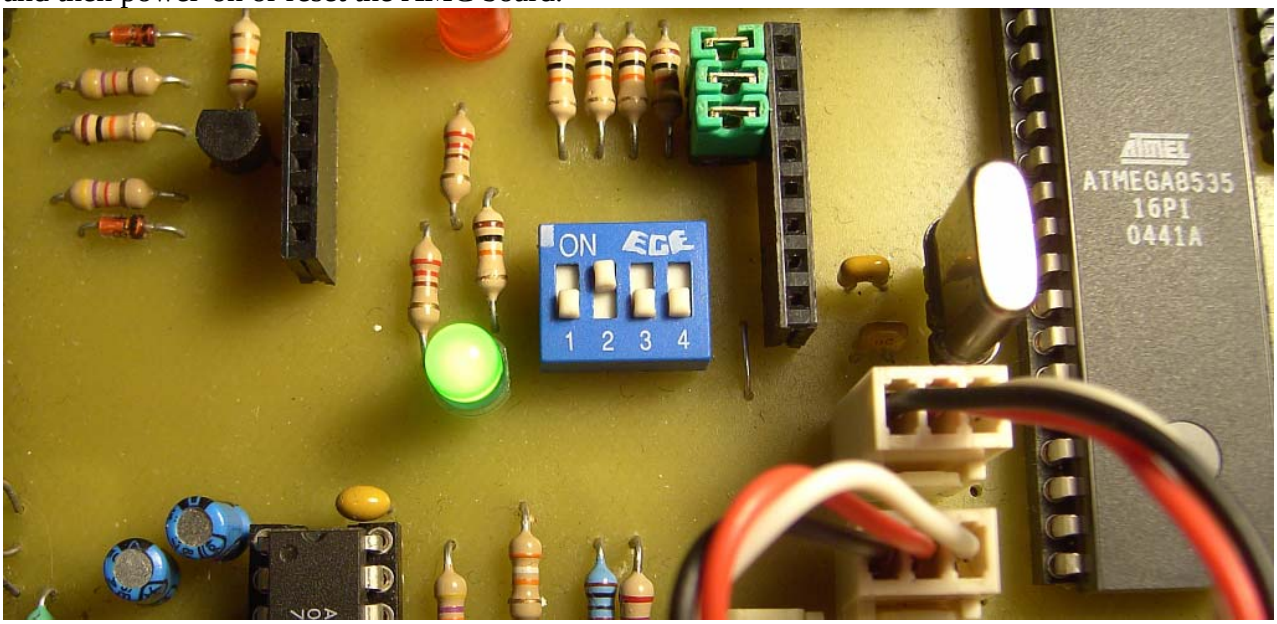
PWM_AMC15_2DOF_V1_CRYSTAL14.HEX (with a 14.31818 MHz crystal)
PWM_AMC15_2DOF_V1_CRYSTAL18.HEX (with a 18.432 MHz crystal)
PWM_AMC15_2DOF_V1_CRYSTAL22.HEX (with a 22.1184 MHz crystal)
PWM_AMC15_2DOF_V1_CRYSTAL24.HEX (with a 24Mhz crystal)

LCD data enable:
This firmware supports display of the motion input data, as well the potentiometer feedback reading, on the LCD. For this purpose you can set the 1st DIP switch to ON position any time (even during the normal operation) to enable the display of data on the first line of the LCD.

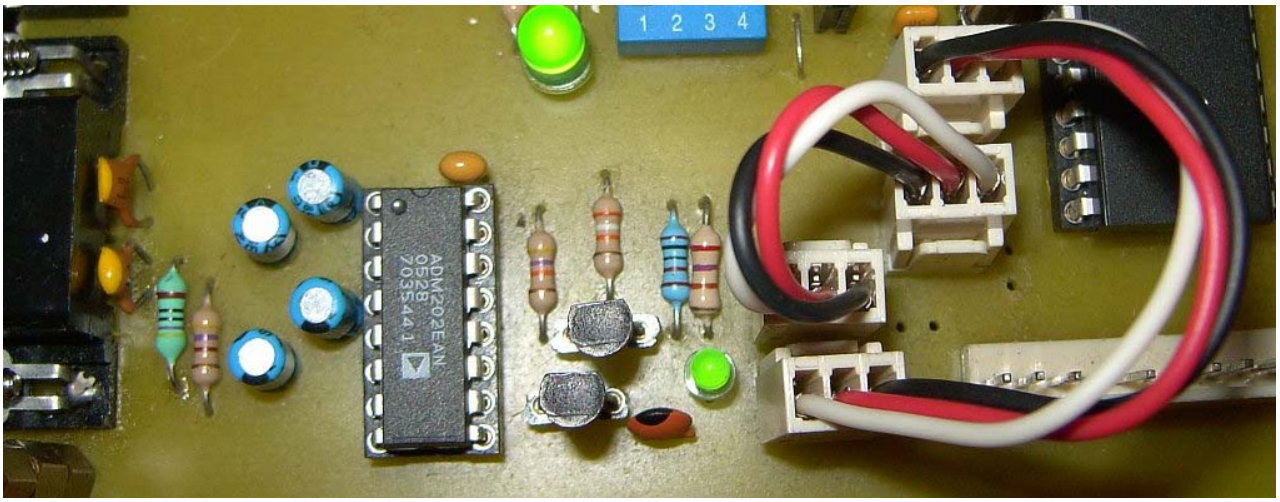


Motion Control parameters:
The motion control scheme of the motors consists of Proportional servo code that controls the acceleration and the positioning of the motor. In many cases the default setting of 3 times the proportionality constant (Kp) might not be enough to give fast motion, or even reaching the destination position. For this reason you must increase the Kp until you find the motion of the motor fast and precise in positioning. Just be careful increase one at a time the Kp values as they are multiplied by the error factor and might end up with too little servo stabilization area and end-up having your motor overshooting the destination position and bouncing forward and backward in a try to position itself. This unbalance can be disastrous if you let it run this way for a while, since each overshooting can be bigger from the previous and it can produce mechanical problems. Depending the type of simulator you may need to assign different Kp parameters for each motor. In my Joyrider I use Kp=7 for the Y-axis and Kp=5 for the X-axis, since the travel of each axis is a little asymmetrical.

To manually set the Kp setting for each motors you have to set the 2nd DIP switch to ON position and then power-on or reset the AMC board.



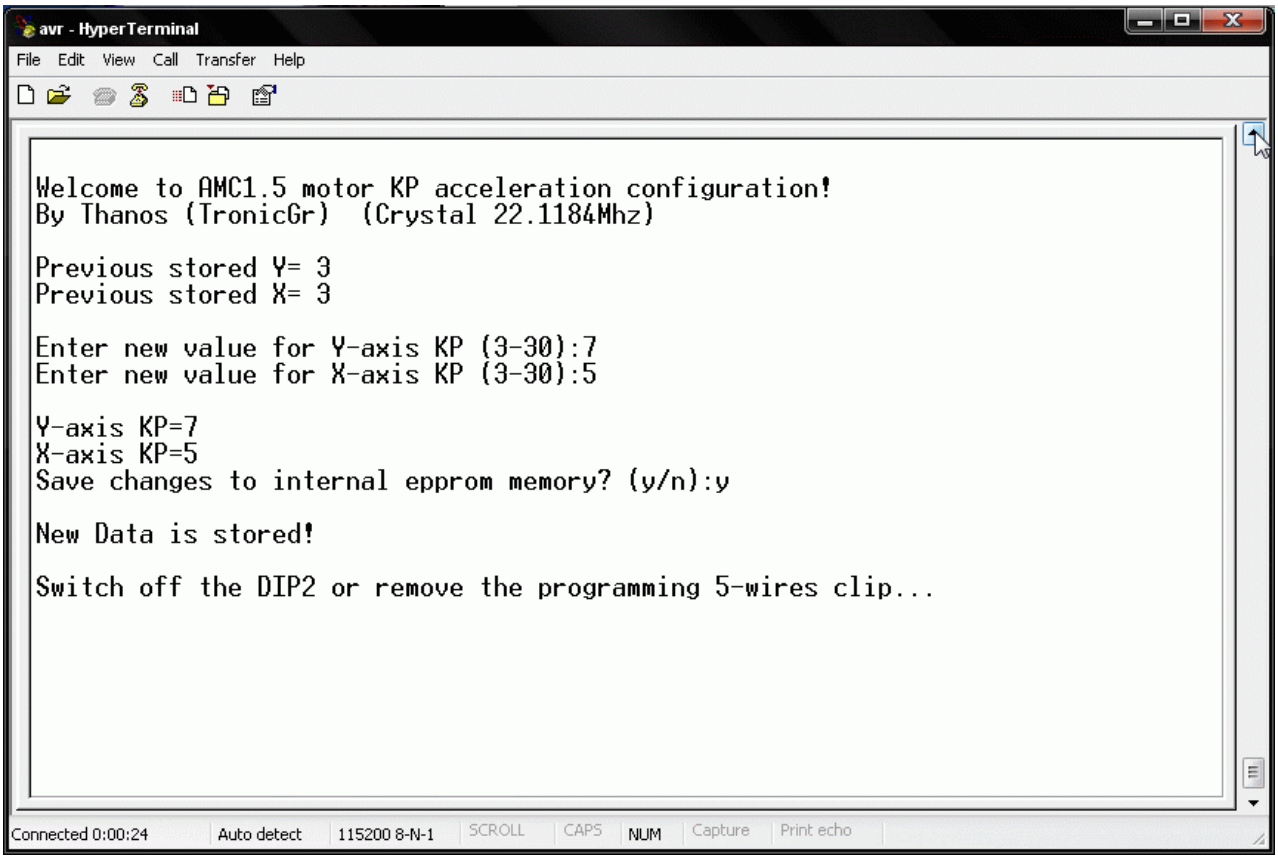
Since the setup of Kp parameters require that the AMC has 2-way communication, you have to also use a second molex cable to connect the two “TX” connectors:



HyperTerminal use:

For entering the Kp parameters of the motors you have to use the windows HyperTerminal to have access in the simple user interface of AMC. This terminal user interface uses the same com port on the AMC, which is used for receiving motion data. No more need to flash different firmware to change the Kp parameters!

Here is an example screen shot:



The steps to start and use the Windows HyperTerminal are:

1. Open HyperTerminal.
2. Enter a name for the terminal window (I called it “AVR”)
3. Select the COM port you are using
4. Set the bits per second as 115200 and Flow Control to “None”
5. From Menu, select View, Font, and increase the system font to size 12.
6. Now set the DIP2 switch to ON position and power-on the AMC1.5 board.

The following will appear on the terminal:

Welcome to AMC1.5 motor KP acceleration configuration!
By Thanos (TronicGr) (Crystal 22.1184Mhz)

Previous stored Y= 3
Previous stored X= 3

Enter new value for Y-axis KP (3-30):

Enter the Kp parameters you need for each axis motor by typing the number and then pressing enter. It will accept values from 3 to 30.

TIP: For SimForceGT motion its recommended to use the same Kp parameter values for both motors or else the motion might not be synchronized! But you can "repair" a slow motor in SimForceGT motion simulator by increasing a little its Kp until it matches the faster motor!

```
Enter new value for Y-axis KP (3-30):7
Enter new value for X-axis KP (3-30):5

Y-axis KP=7
X-axis KP=5
Save changes to internal epprom memory? (y/n):
```

It will then ask if you want to save the changes you made in the internal epprom memory so they can be used each time you power the AMC automatically… You can press "y" or "n" here, and press Enter.

If you press "y" this will show on the terminal:

```
New Data is stored!
```

In this case the parameters are written in the memory.

If you press "n" this will show on the terminal:

```
Procedure is aborted!
```

In this case nothing will be written in memory.

Another message will also show that indicates you switch off the DIP2 to continue the normal operation of the AMC. It will also suggest to remove the 5-wire programming clip cause the circuit of the ISP programmer pulls up the DIP2 data line if left connected resulting in leading the AMC into setup mode each time it power up:

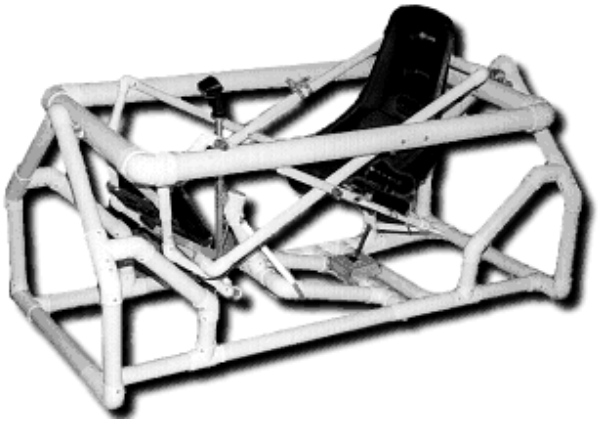
```
Switch off the DIP2 or remove the programming 5-wires clip...
```

Note: The first time that you run this firmware you will see that the Kp values for both Y,X axis are 255 (FF). This is normal. If you cycle the power or reset the AMC board, an internal safety check I installed will save automatically the default values of Kp=3 for both Y,X axis. This is the same internal safety check that prevent you from entering values smaller than 3 of bigger than 30 as Kp parameters as they are the absolute limits inside that the software can operate.

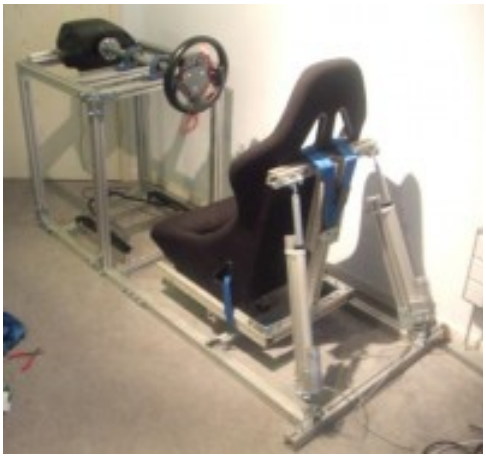
Types of motion simulators supported by this firmware:

2DOF uncoupled (JoyR), like the motion of Joyrider
2DOF coupled (SFGT), like the motion of SimForceGT (also known as frex-style)

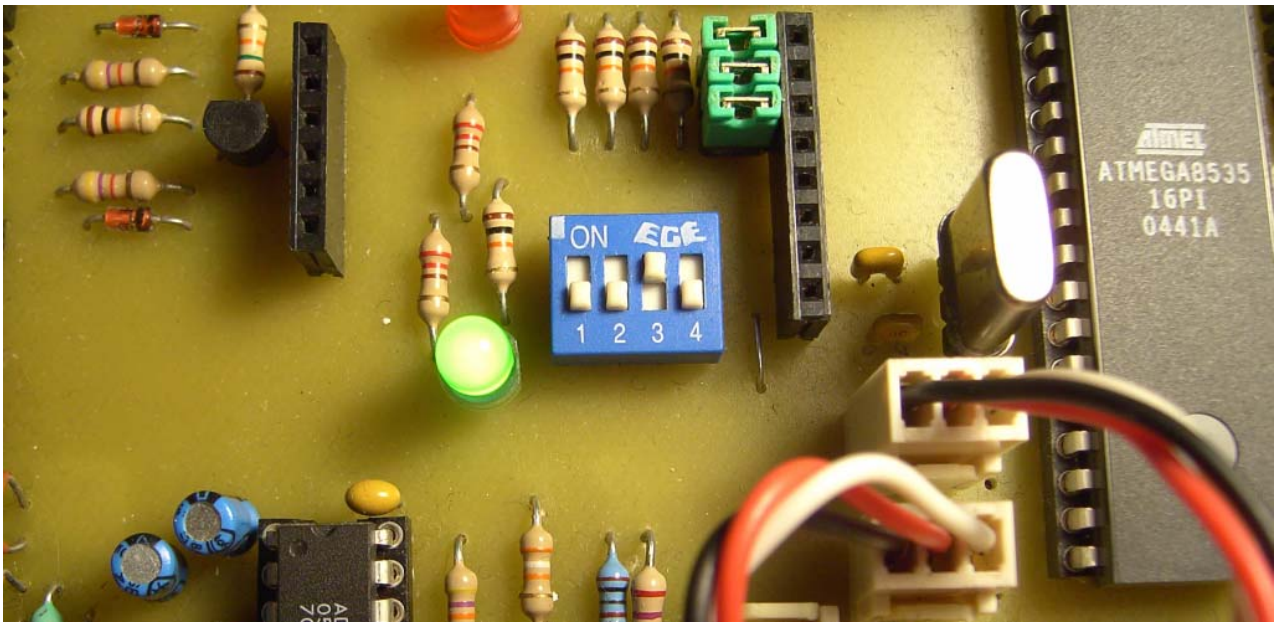
Here are some Joyrider motion simulator type examples:



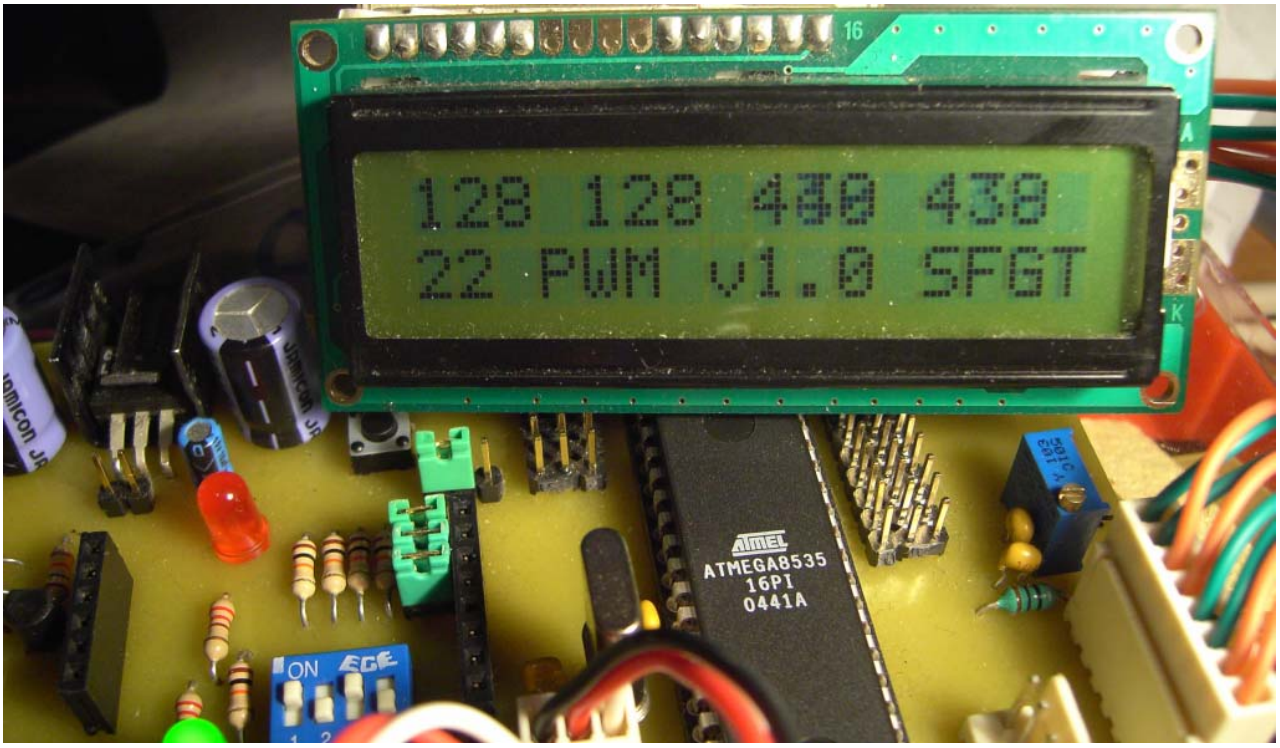
Here are some SimforceGT motion simulator examples (Egoexpress):



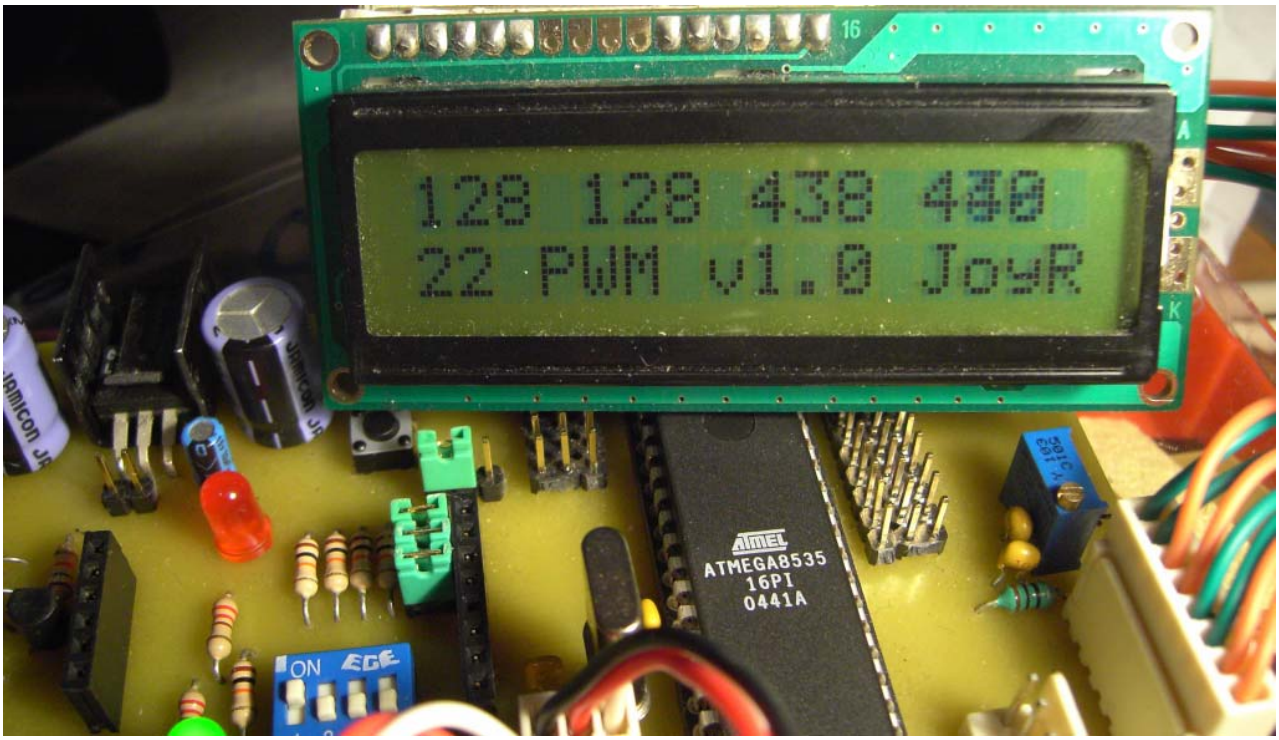
With this firmware you can change the type of motion style using the 3rd DIP switch:
OFF = JoyR
ON = SFGT



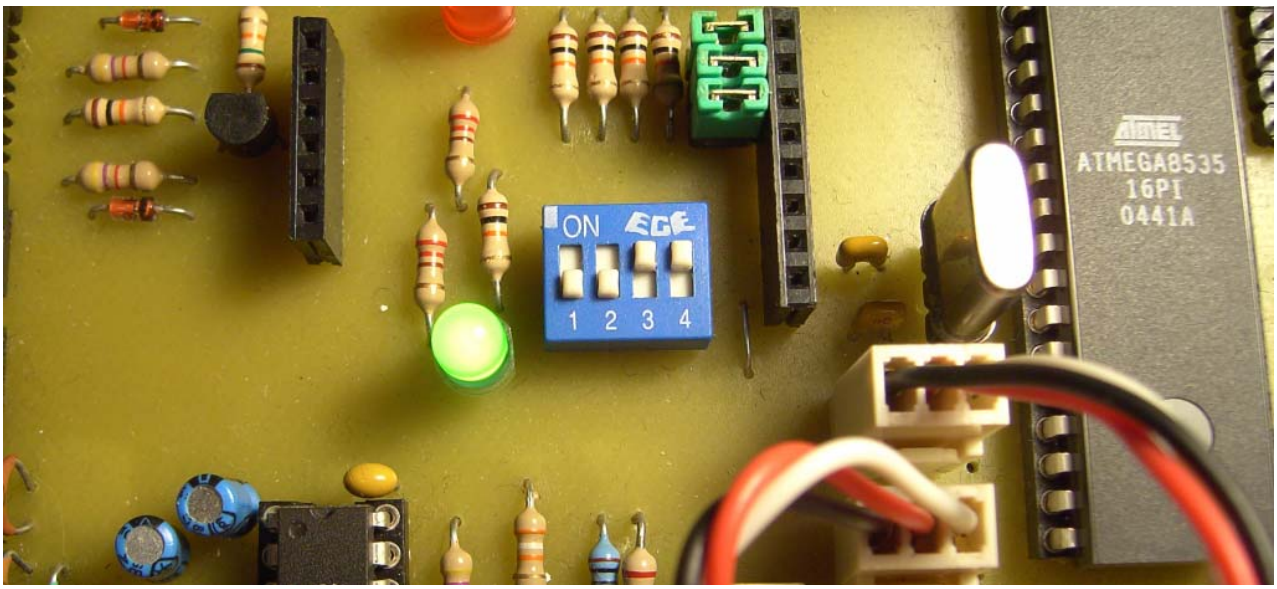
And here is shown the version of the firmware along with the crystal speed and the motion type (SFGT motion style is shown here):



And here is shown the JoyR mode (Joyrider uncoupled type simulator):



In some occasions it’s necessary to swap the motors while in SFGT motion style to correct the direction of the motion but swapping of the physical connections might result the wrong axis coupling! In this case you can set the 4th DIP switch to ON position to swap the software mixing of the motion coupling.
OFF = Normal software motion coupling
ON = Swapped software motion coupling



By Thanos (TronicGr) Greece, Athens 2008