**MOTORCONTROLL**

This is an application which can control 3 DC-motors individually and change between to different positions.

**Getting Started**

* This is a terminal program and all commands take effect through the terminal.
* When the user writes the command ‘INJECT X’ will motorX rotate in a forward direction in a specific amount of milisec.
* When the user writes the command ‘EJECT X’ will motorX rotate in a reverse direction in a specific amount of milisec.
* Parameters:
  + Every motor has a set with parameters (all are integers):
    - Time the motor will run forward (in milisec, 0-3000): ‘TFWD’
    - Time the motor will run reverse (in milisec, 0-3000): ‘TREV’
    - Speed in forward direction (in%, 0 - 100): ‘SFWD’
    - Speed in reverse direction (in%, 0 - 100): ‘SREV’
  + Every parameter is customizable in the command line. Syntax (with TFWD as example):

**SETx TFWD=y => Will set TFWD for motor X to y msec**

**Every parameter is readable from the cmdLine:**

**GETx TWFD => Will return ‘TFWD=y’**

**States**

* Every motor has 4 states. It can vary between ‘EJECTING’, ‘INSERTING’, ‘EJECTED’,’INSERTED’.
* When the user types string ‘STATEX’, will the application reply with the state motor X is in.
* Application always starts in state ‘EJECTED’.

**Retur values**

* All commands which the application accepts will return ‘200 OK\r\n’.
* All commands which the application does not accept will return ‘400 Error\r\n\’.

**Terminal**

* Everything that is sent to the terminal will be echoed back.
* Backspace works as inteneded.
* The Delete-button deletes the current string.

**Prerequisites**

STK600 must be powered through the USB connection. For the application to work the UART connection through RS232 must be connected (Jtag is optional for debugging). In addition, the user must change some settings on the STK600 to power the ATMEGA2560 and set the correct frequency. ATmega2560 uses 5v and 8Mhz programable clock, and can be set from ATMEL Studio 7:

**Download all files from git.**

**Open the project in Atmel studio**

**Tools-->Device Programmer**

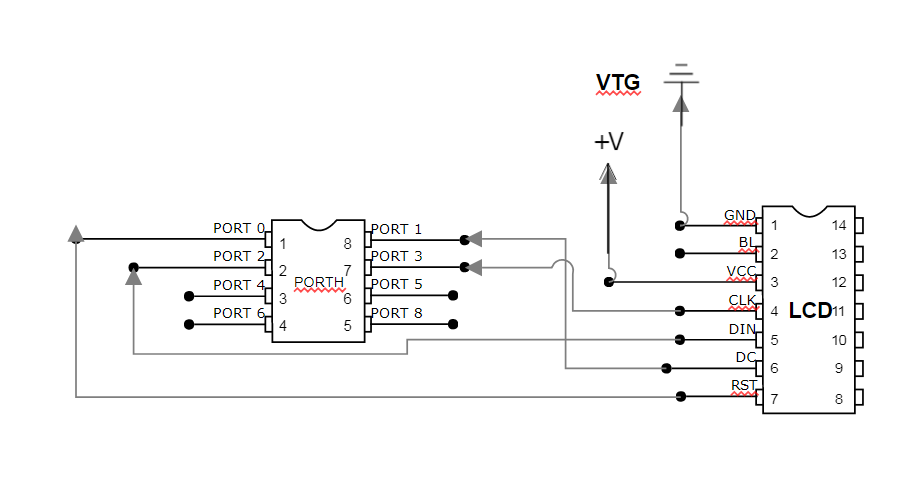
**From the pulldown menu select "STK600", Apply--->Read-->Board settings**

**"SET Vtarget to 5v and clock generator to 8Mhz"**

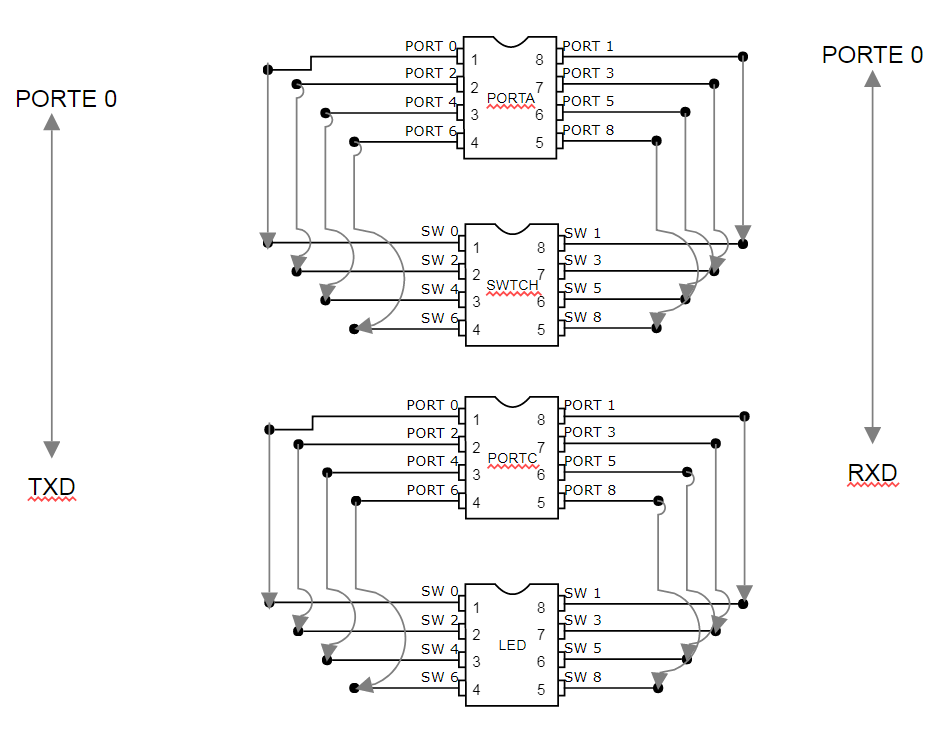
**Installing**

This is a step by step series of examples which will explain the user how to set up the application. By following the instructions, which the pictures bellow illustrates, can the user connect the LCD, motor, and general interconnections on the STK600(Datasheet for more information)

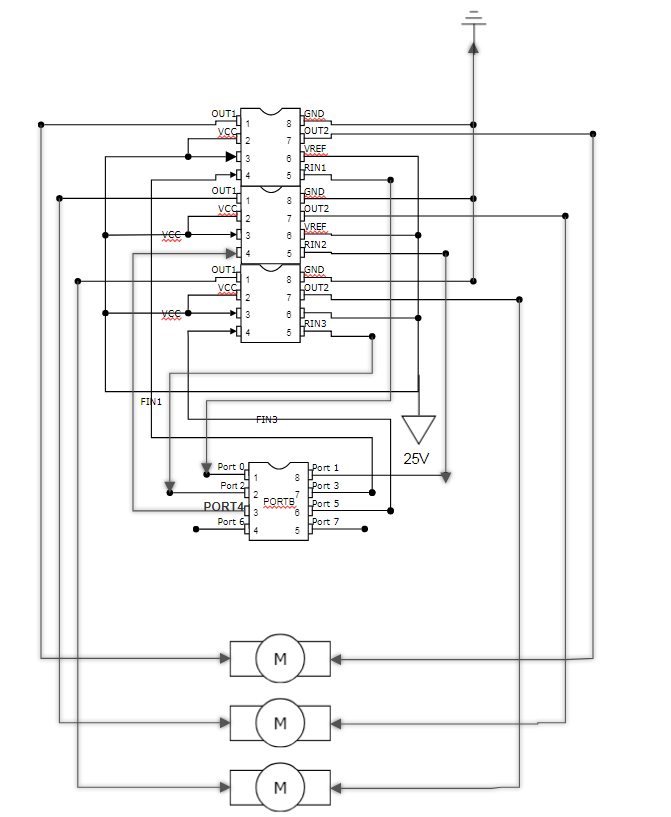
**LCD-STK600 connections(Nokia5110):**

[](https://github.com/HenriknWold/WindowMote/blob/master/MOTORKONTROLL/LCD_NOKIA.PNG)

**Interconnections(STK600):**

[](https://github.com/HenriknWold/WindowMote/blob/master/MOTORKONTROLL/INTERCONNECTIONS.PNG)

**MotorControll to STK600(BD6230F):**

[](https://github.com/HenriknWold/WindowMote/blob/master/MOTORKONTROLL/MOTOR.PNG)

* Make sure that the there is one common connection to ground.
* Make sure that STK600 is preconnected correctly (Datasheet STK600)

**Test the application**

Sett DC Power supply to 10-25V. (25 optimal)

**PRESS SW5 UNTIL NUMBER ON THE LEDLIGHT REPRESENTATES THE RIGHT MOTOR THE USER WHATS TO RUN MANUALY.**

**HOLD SW6 OR SW7 UNTIL THE MOTOREN IS MOVED TO THE CORRECT POSITON (EJECTED POSITION)**

The user is now ready to test the terminal application.

Start with opening TeraTerm or any other terminal program. Write the following command and make the correct observations.

**INJECT "X"(X stands for the motor number)**

\*Observe that the motor rotates clockwise for 3seconds at a medium velocity. \*Observe on the LCD screen that the state of MX has been updated to "INSERTED".

For further testing type the following cmd:

**EJECT X**

**STATE X**

**SETX SFWD=80**

**SETX SREV=80**

**GETX SFWD**

**GETX SREV**

Now the motor should rotate anti clockwise and respond in the terminal window with **"EJECTED", SFWD=80, SREV=80**.

This means that the user has now changed the speed of the motor in forward and reverse direction from 50% to 80%. In addition, the user can now observe that the LCD has updated it states again.

**Datasheet**

* [STK 600](http://ww1.microchip.com/downloads/en/DeviceDoc/40001904A.pdf)
* [ATMEGA 2560](http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-2549-8-bit-AVR-Microcontroller-ATmega640-1280-1281-2560-2561_datasheet.pdf)
* [DC Brush Motor Drivers(36VMax) BD623xxx Series](http://www.farnell.com/datasheets/2097985.pdf)
* [Nokia5110 LCD(48 × 84 pixels matrix)](https://www.sparkfun.com/datasheets/LCD/Monochrome/Nokia5110.pdf)

**Built With**

* [ATMEL Studio 7](https://www.microchip.com/webdoc/GUID-ECD8A826-B1DA-44FC-BE0B-5A53418A47BD/index.html?GUID-8F63ECC8-08B9-4CCD-85EF-88D30AC06499) - is the integrated development platform (IDP) for developing and debugging used for this application
* [TeraTerm](https://ttssh2.osdn.jp/index.html.en) - terminal emulator (communications) program.