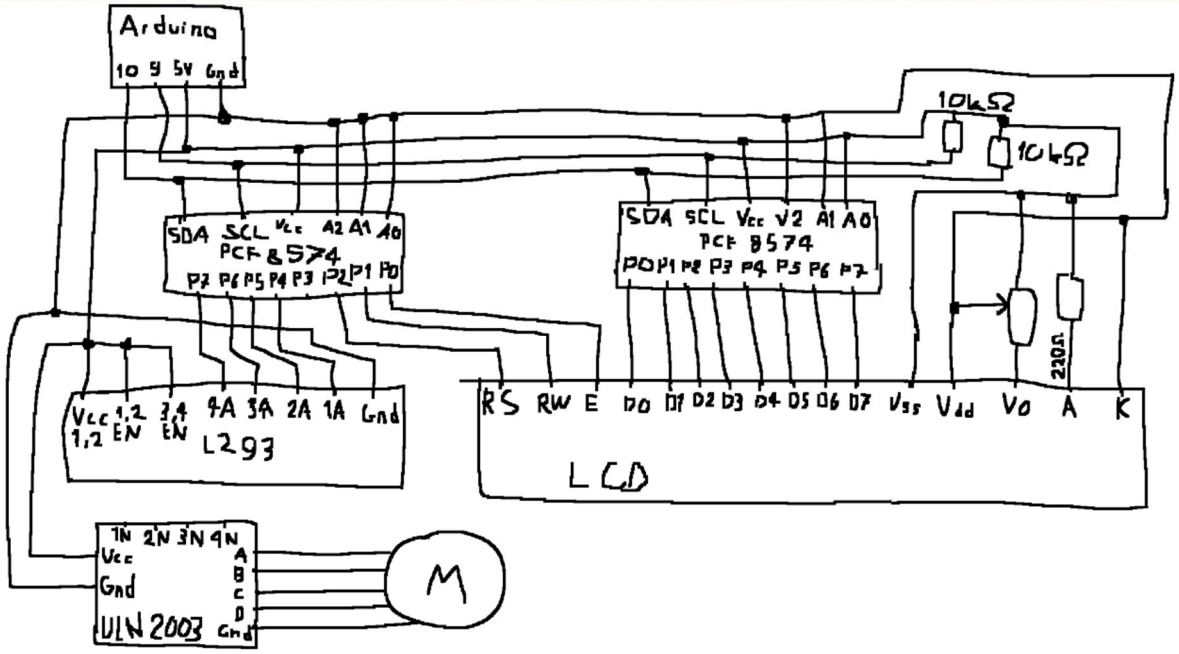
Stepper motor controller

Project by: Zbigniew Barczyk

The Circuit designed uses two I2C expander chips – PCF8574a to control an LCD display and a stepper motor. The I2C chips are connected to the Arduino with two wires: a data wire and a clock wire. Each I2C chip has eight data pins that we can use. The LCD display uses 8 bits from one of the I2C chips – D0 to D7, as well as 3 pins from the second I2C chip – RS, RW and EN. The stepper motor only needs four wires from the second chip leaving one wire open and unused. The data and clock wires connecting the I2C chips with the Arduino are connected to 10k ohm pullup resistors to set initial state to high and remove noise and unwanted data. On top of this the data outputs from the second expander chip controlling the stepper motor are connected via a L293d transistor array to increase the voltage to be sufficient for the ULN2003 stepper motor drive module to power the stepper motor.

The stepper motor and LCD display are controlled using a custom library for the I2C chips. The library has all the functions with the correct timings to set the LCD display on and display provided words. It also contains functions for controlling the stepper motor with the ability to change speed.



The function writeData(char) sets the given value as the data to write, similarly writeCmd(char) writes a command to the display. These two commands are used as a base for all the other commands such the ones for clearing the

display, changing to second row, or writing text on the display. WriteCharArray function displays the two words it is provided with on the LCD display. Function write1Byte is used in the function write1Message. Together they send 1 byte bitwise to the correct I2C chip with the correct timing and acknowledgement. Functions startSignal, stopSignal and readAck have the timing and pin mode to start sending signal, stop sending signal and receive the acknowledgement. In this library we ignore situations where we don't receive the acknowledgement. Functions clockwise and anticlockwise are responsible for setting the correct pins on the second I2C chip to high and low to control the stepper motor clockwise and anticlockwise.

The library works fine with the code. It correctly controls and displays the state of the motor and lcd. It allows to use two wires to control 15 data pins. The stepper motor turns clockwise and anticlockwise at different speeds and the LCD display correctly displays the state of the stepper motor. However, the library code is a bit messy. It contains inconsistencies regarding the addresses and code. Some functions use hexadecimal values where else other use binary values. This is not an issue, but it would be nice to have more consistency in the code. The program for the Arduino itself is very simple and straight forward thanks to the custom library that was designed perfectly for this task.