# Summary

CNCLib can be used for CNC machines.

The first part of the project implements a library to customize CNC machines on different Arduino hardware.

The second part is designed for Windows to communicate with the CNC (Arduino) machine.

# Arduino CNC

The library supports a big range of Arduino hardware

8bit

* Arduino Uno
* Arduino Duemilanove
* Arduino Nano
* Arduino Pro and Pro Mini
* Arduino Mega 2560

32 bit

* Arduino Due
* Arduino Zero
* Arduino M0

Depending on your hardware different stepper implementations can be used:

* Gbrl compatible, e.g. CNCShield, … running on Uno (8bit) or Zero(32bit)
* Ramps 1.4  
  The Arduino Mega 2560 shield (designed for 3D printers) can be used to create a CNC machine with LCD, SD and up to 5 axis. The GCode parser is extended to support more codes and also 3D rotations of the coordinate system is possible.  
  (with some restrictions you can use the Arduino Due to improve performance)
* Ramps FD  
  This is a shield for the Arduino Mega 2560 (8bit) or Arduino Due (32bit). The shield supports 6 axis and all other features of the Ramps 1.4
* Generic Step/Dir Steppers.   
  The library includes a sample how to create a CNC machine without using a shield.  
  In the sketch KK1000.INO a naked Due is used to connect a CNC machine with a DSUB25 connector (with a 3.3 to 5V level converter). A LCD (from Ramps 1.4) is connected which also gives the machine SD support.
* Generic Servo Support  
  Servos aren’t used in CNC machines. But servos can be used for e.g. a robotic arm. The example sketch iRobot.ino converts the cartesian coordinate system to the 3(+1) angles of a servo driven robotic arm.  
  The hardware do not use any shields but a ramps 1.4 LCD is connected (LCD, SD).
* H-Bridge  
  The H-Bridge L298N or ULN2003 (or equivalent) are supported by the library.
* SMC800 Stepper driver
* The library can be extended to use other stepper implementations. C++ code necessary.

## Supported GCodes

|  |  |  |
| --- | --- | --- |
| Code |  | Description |
| G0 | Std. | Additional F (without value) |
| G1 | Std. | Cut Move |
| G2 | Std. | Arc |
| G3 | Std. | Arc |
| G4 | Std. | Dweel |
| … |  |  |

## Installation

Uploading the CNC program to the Arduino can be done in two different ways:

1. Upload a hex file.  
   The hex file can be downloaded from the CNCLib Web project site (github, SourceForge). To upload the file to the Arduino you can use e.g. the tool Xloader (see <http://www.hobbytronics.co.uk/arduino-xloader> )
2. Arduino IDE  
   If no hex file is available or the sample-sketch must be changed, the Arduino IDE (1.8.x) must/can be used. Please note, the Arduino program consists of 3 libraries. The library file must be copied to the Arduino library path or it is also possible to change the “Sketchbook location”.  
   Please specify the   
    Stepper/Sketch  
   folder in the downloaded and extracted CNCLib folder.

## Examples

### MiniCNC

The sample is designed to run on a small Arduino (Atmel328p). Because of the restricted hardware no LCD and SD is used. In the default configuration the grbl 1.1 PIN assignment is used.

Upload the unchanged sketch and modify the CNC machine parameters. This can be done by using the Windows CNCLib application.

### MiniLaser

The sketch is written to support a DIY laser engraving machine. No end stops are used and the PIN configuration is a modified version of grbl 1.1. Please see the sketch files to see how to swap the “dir” and “step” pin.

### Laser

The sketch is a preconfigured version of “MiniCNC”. The DIY laser is built with a CNC Shield (grbl 1.1 PIN Layout with a PWM laser on PIN 11).

### DCK40Laser

The “upgraded” version of the DC-K40 china laser is implemented in his sketch. The hardware update of the laser is an Arduino Due with a Ramps FD shield (LCD and SD supported).

The sketch also shows how to extend the library. For example the behaviour of G0 (laser off) and G1 (laser on) is changed in this sketch. Unused IO pins (the shield is used for 3D printers) are connected to a relais turning the water pump and the vacuum on and off.

The parameters of the sketch must be configured at compile (upload) time. The Arduino Due has no Eeprom to store modified configurations.

## Installation

# Windows

# CNCJoystick

All replies from the CNCJoystick starts with ;

If the replies are sent to the CNC machine, it shouldn’t cause anything because this is a comment

* Reply after sketch startup:

;CNCJoystick, Jan 2 2017

* Reply after an Error:

;ERROR

* Writing configurations values:

maxspeedfast in mm/min, default 3000

maxspeedslow in mm/min, default 400

interval sending interval of analog stick, in ms, default 200

count assign a value will reset counters

Examples

maxspeedfast=500 set the maxspeed to 500 mm/min

* Reading values:

Button 1: ;btn1:0

Button 2: ;btn2:0

Button 1 with shift-key: ;btn1s:0

Button 1 with control-key: ;btn1c:0

Button 1 with shift & control-key: ;btn1sc:0

The number after the colon represents the press count of the button.

Please remark, the counts are distinguished between shift, control , …

* Analog

X/Y

Z/A with shift-key