miniEngine v2.0 alpha

Documentation

Software version 2.0 alpha 0
Hardware revision D

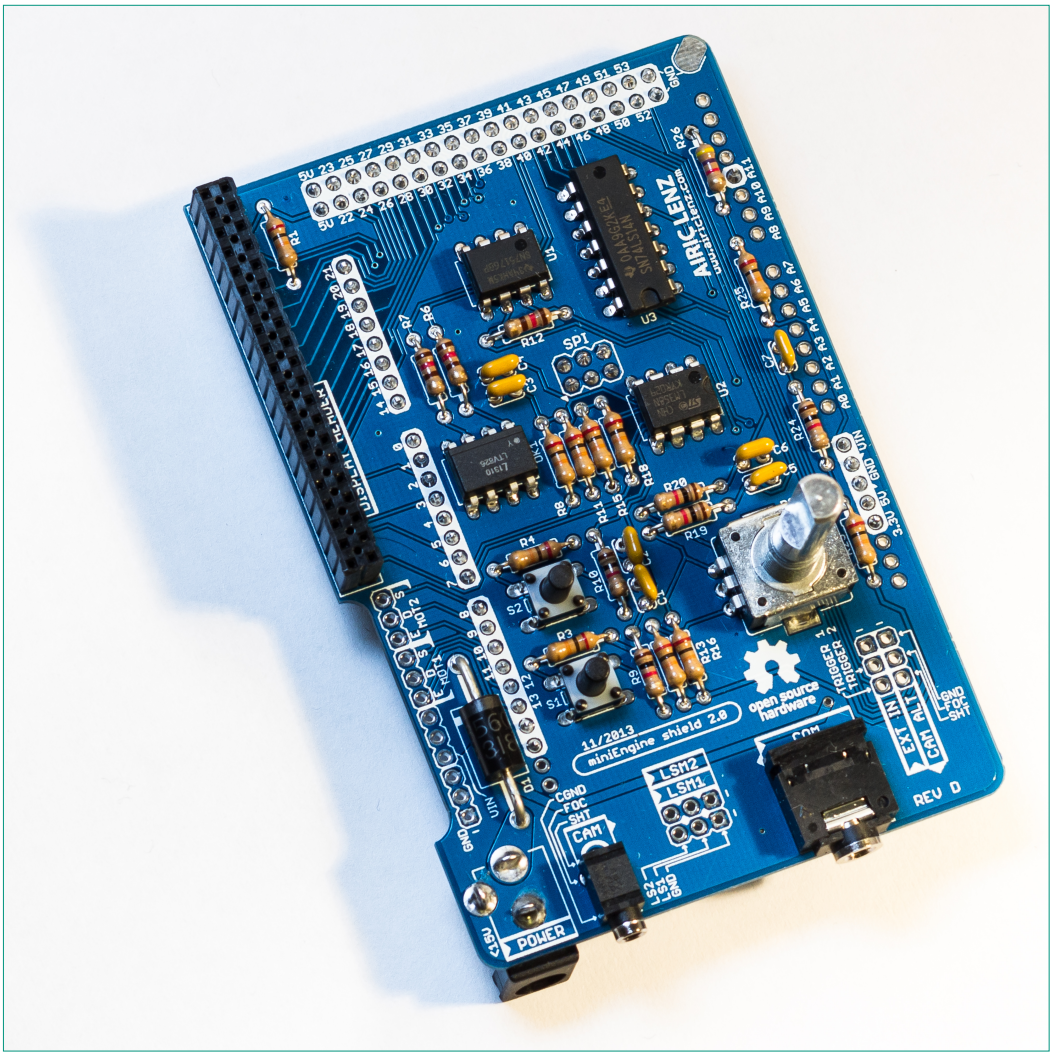
Airic Lenz

Content

1 | Introduction

The miniEngine is open-source motion-control-system for time-lapse as well as video photography. The current version provides all functions needed to smoothly and precisely control 2 stepper-motors and one camera. It is Arduino DUE based and comes with a shield, designed to hold all required hardware components. These components allow user input as well as connecting 2 stepper-motor-drivers for then connecting the motors.

This is how a fully assembled miniE (short for miniEngine) shield for the Arduino DUE looks like (the display is not plugged into the shield for a better view):



The miniEngine 2 Arduino shield

These function are currently supported by the system:

- Simultaneous control of two stepper motors
- Control of one photo-camera (needs to be remote trigger-able)
- Time-lapse recording in shoot-move-shoot mode
- Time-lapse recording in continuous mode (also usable for video shoots)
- System is fully adjustable to the hardware used, e.g. stepper motor calibration to be able to set up precise moves in cm / °
- User input via one rotary encoder and two buttons
- 320 x 240 pixel colour TFT display
- All settings / program data stored on an SD card for easy and fast reconfiguration using multiple cards holding different set-ups
- Fully open-source

These are the major components that are required for the system:

1	Arduino DUE with the miniEngine 2 software installed
1	ITDB02-2.4E Display (320x240 Pixel, SD-Card, Touch) from ITEAD Studio
1	miniEngine 2 Arduino shield (can be bought on www.airiclenz.com)
1 or 2	Stepper Motor Drivers (BigEasyDriver is recommended and supported)
1 or 2	Stepper Motors
1	7-15V Power supply capable of supplying up to 4A

2 Hardware (miniEngine 2 Arduino Shield)

The following chapter explains how to assemble the miniEngine shield and which components are needed in detail.

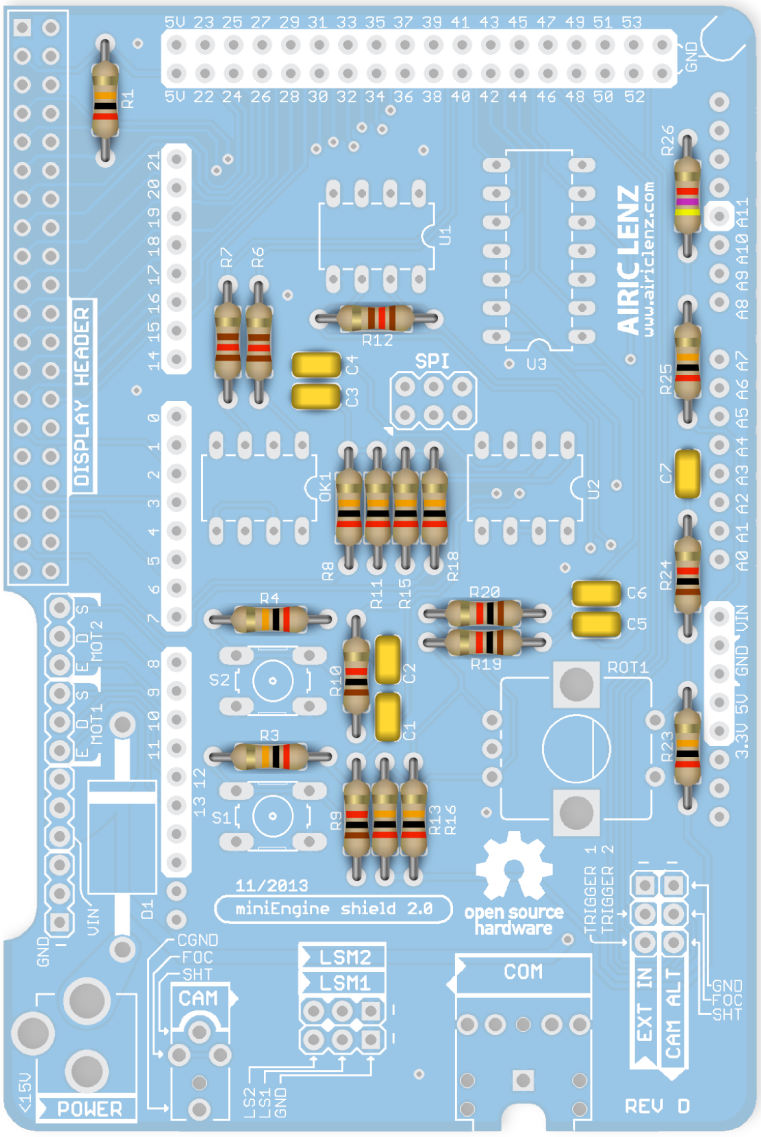
There is an additional power input on the miniEngine shield. This input is secured up to 5A by a beefy diode and can easily handle the amount of energy the stepper motors might need. USE THIS POWER-JACK and NOT the original power-jack of the Arduino!!! Otherwise you might destroy the Arduino!

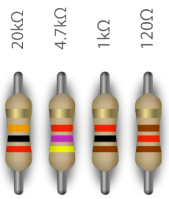
2.1 *Bill of Material*

Amount and Part	Board designator(s)	Description / package
11x 20kΩ Resistor	R1, R3, R4, R8, R11,R13, R15, R16, R18, R23, R25	1/4W, 5%, THT Resistor
5x 1kΩ Resistor	R9, R10, R19, R20, R24	1/4W, 5%, THT Resistor
3x 120Ω Resistor	R6, R7, R12	1/4W, 5%, THT Resistor
1x 4.7kΩ Resistor	R26	1/4W, 5%, THT Resistor
7x 100nF Capacitor	C1, C2, C3, C4, C6, C6, C7	2.54 mm pitch Capacitor
1x 5A Diode	D1	DO201AD
1x 2.5MM Stereo Jack	CAM	2.5mm Headphone jack
1x 3.5MM Stereo Jack	COM	3.5mm Headphone jack
1x 2-Channel Optocoupler	OK1	LTV826, DIP-8
1x Differential Bus Transceiver	U1	SN75176B, DIP-8
1x 2-Channel Op-Amp	U2	LM358, DIP-8
1x Hex Inverting Schmitt Trigger	U3	SN74HC14N, DIP-14
2x Tactile Switch	S1, S2	Tactile Button, THT
2x 1x18 Male Pin Header	Arduino Pins (marked white)	Male Pin Header with 2.54mm pitch, THT

4x 1x8 Male Pin Header	Arduino Pins (marked white)	Male Pin Header with 2.54mm pitch, THT
1x 2x20 Female Pin Header	DISPLAY HEADER	Female Pin Header with 2.54mm pitch, THT
1x 2x3 Female Pin Header	SPI	Female Pin Header with 2.54mm pitch, THT
1x Rotary encoder	ROT1	PEC11, 25mm shaft, 24 steps, 24 indent, switch
2x Limit switch	LSM1, LSM2	Generic 2x3, THT, 2.54mm pitch header
1x Trigger inputs & alternative Camera output	EXT IN, CAM ALT	Generic 2x3, THT, 2.54mm pitch header

| 2.2 | *Assembly Steps*





Step 1

Solder all Resitors and all Capacitors to the board:

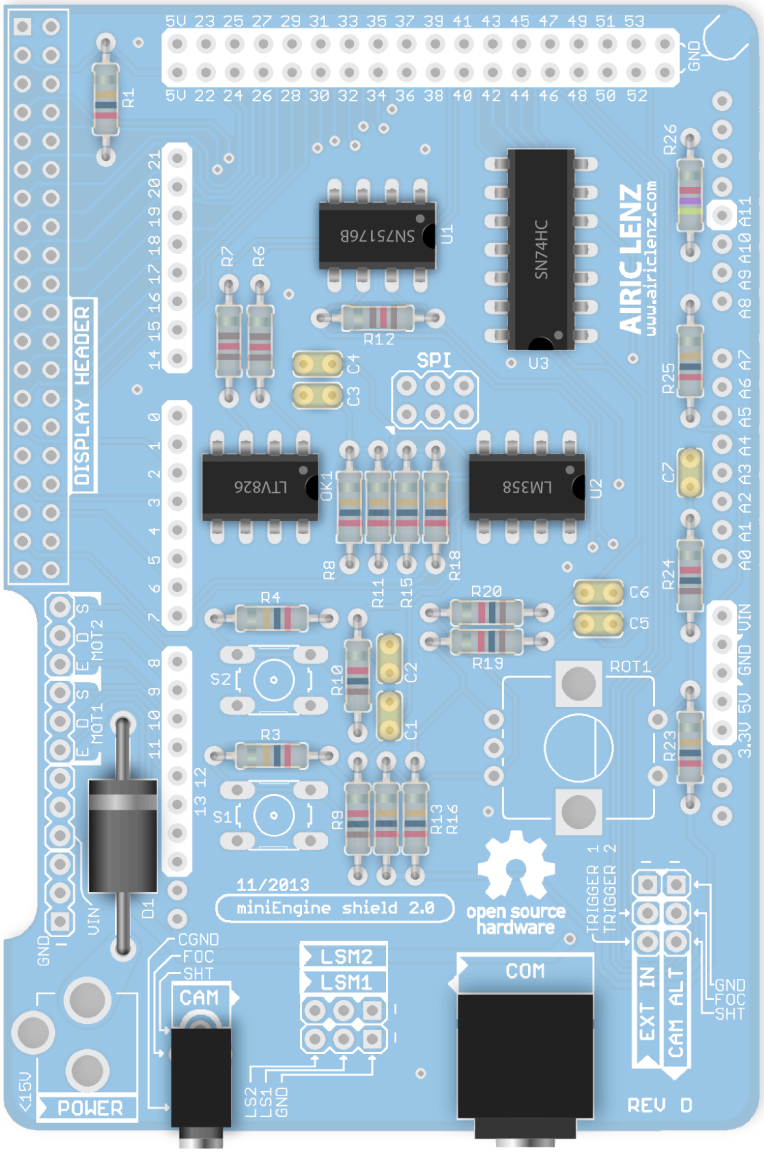
20kΩ: R1, R3, R4, R8, R11,R13, R15, R16, R18, R23, R25

1kΩ: R9, R10, R19, R20, R24

120Ω: R6, R7, R12

4.7kΩ: R26

100nF: C1, C2, C3, C4, C5, C6 ,C7



Step 2

Solder all medium sized top side components:

U1: SN75176B (Communication)

U2: LM358 (Backlight amplifier)

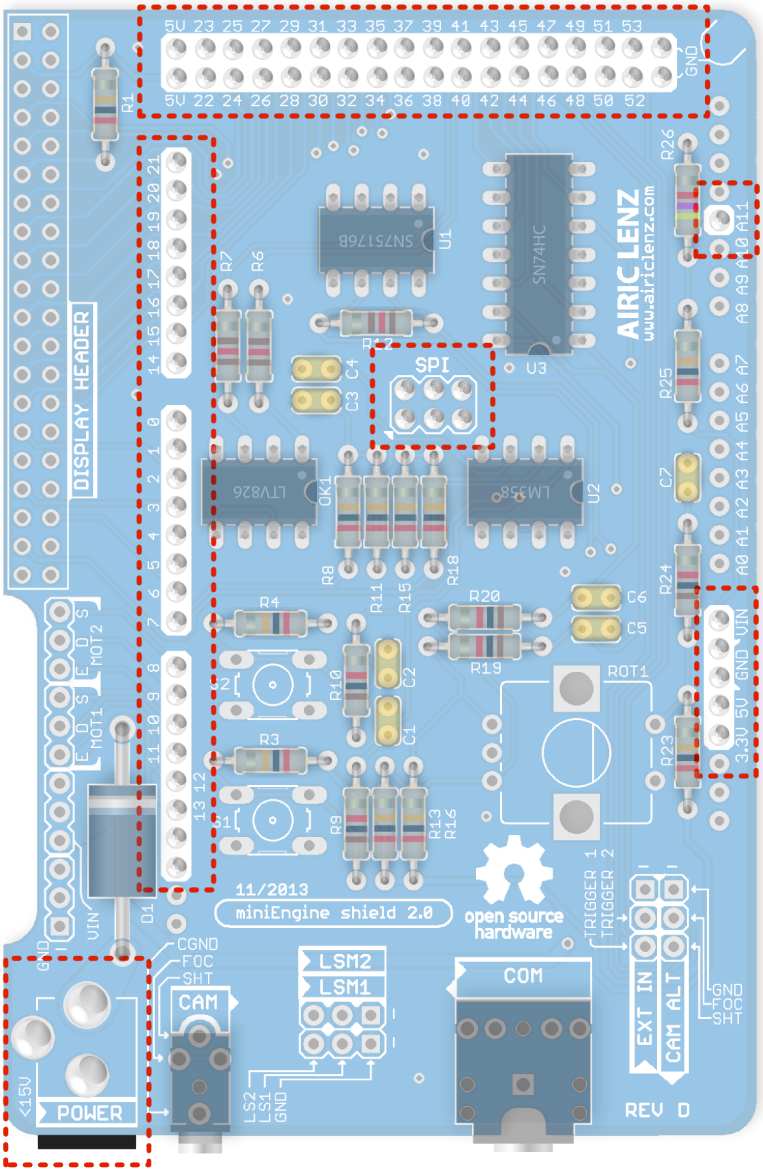
U3: SN74HC (Invert. Schmitt Trigger)

OK1: LTV826 (Optoisolator)

D1: 5A Rectifier Diode

CAM: 2.5 mm Stereo Jack

COM: 3.5 mm Stereo Jack



Step 3

Solder all BOTTOM side components:
MUST BE SOLDERED TO THE BOTTOM SIDE!

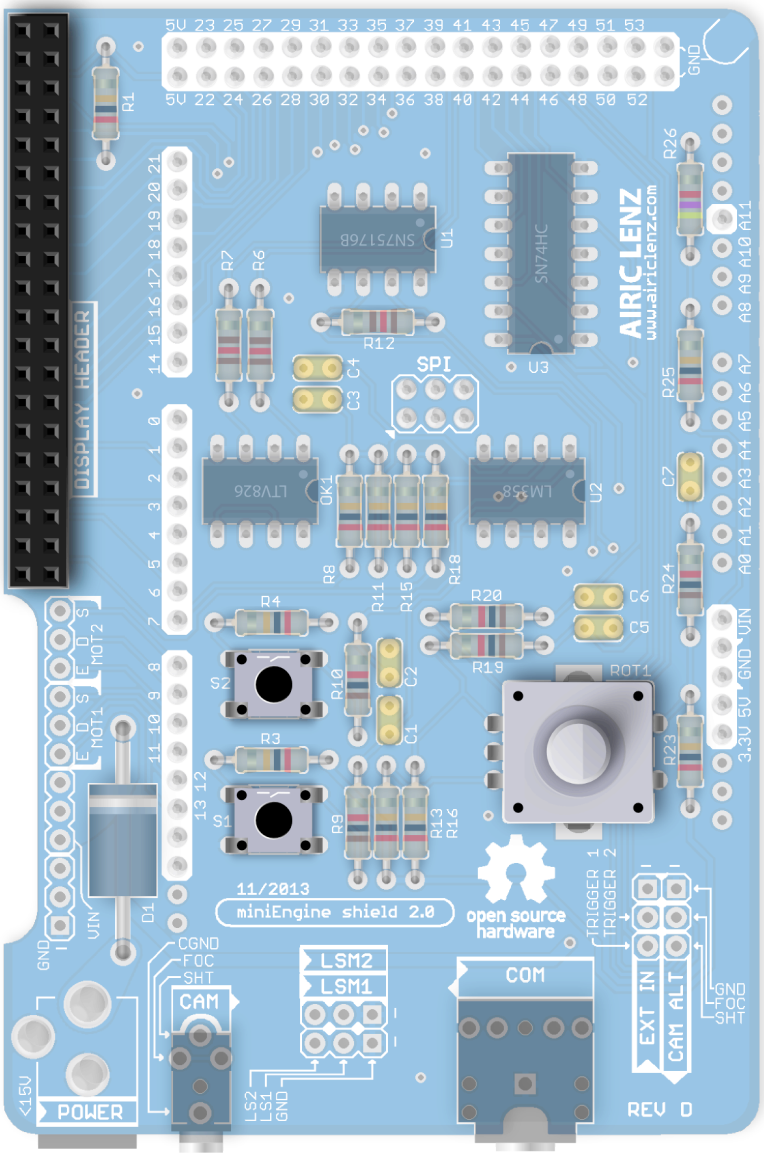
POWER: 2.1mm Power Jack (inner pin positive)

Arduino Pins: Male Pin Headers need to be soldered to the white marked holes! You can solder umarked pins too but white marked ones must be soldered.

4 headers with 8pins (one needs to be broken for Power Pins and Analog input pin 11)

2 headers with 18pins

SPI Port: 3x2 Female Header



Step 4

Solder the big top side components:

S1, S2: Tactike Switches

DISPLAY HEADER: 20x2 Female Header

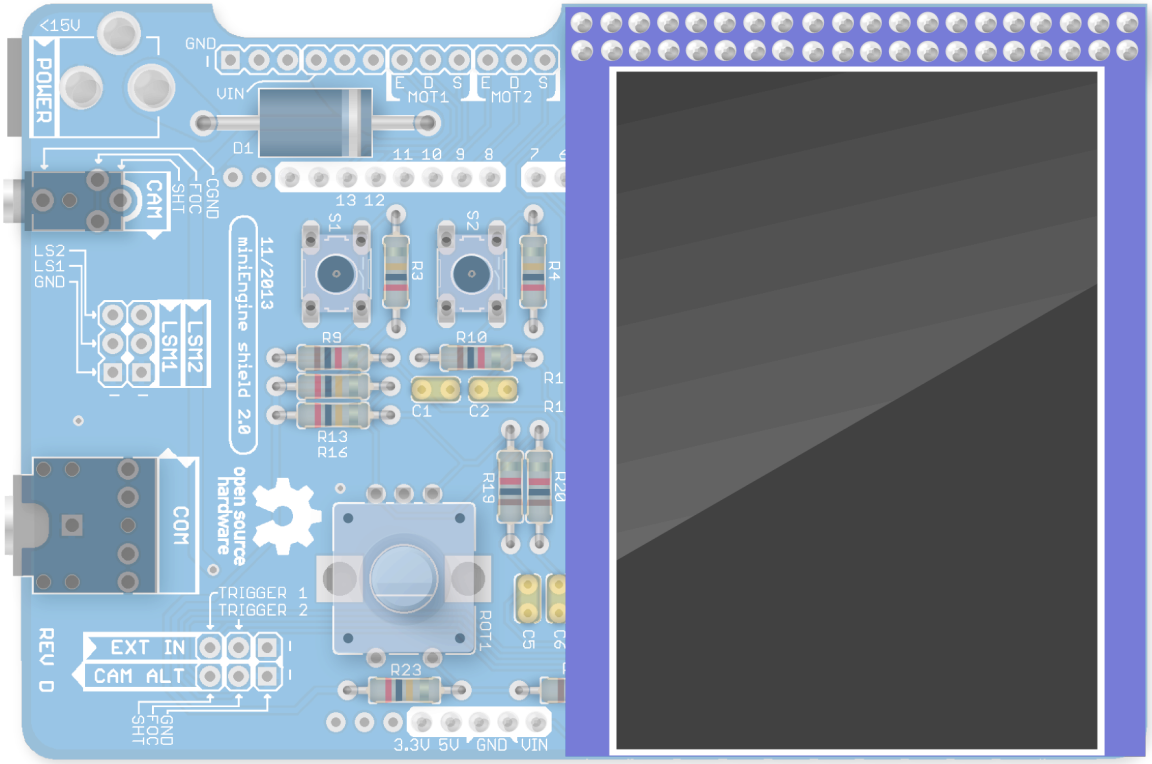
ROT1: Rotary Encoder

Step 5

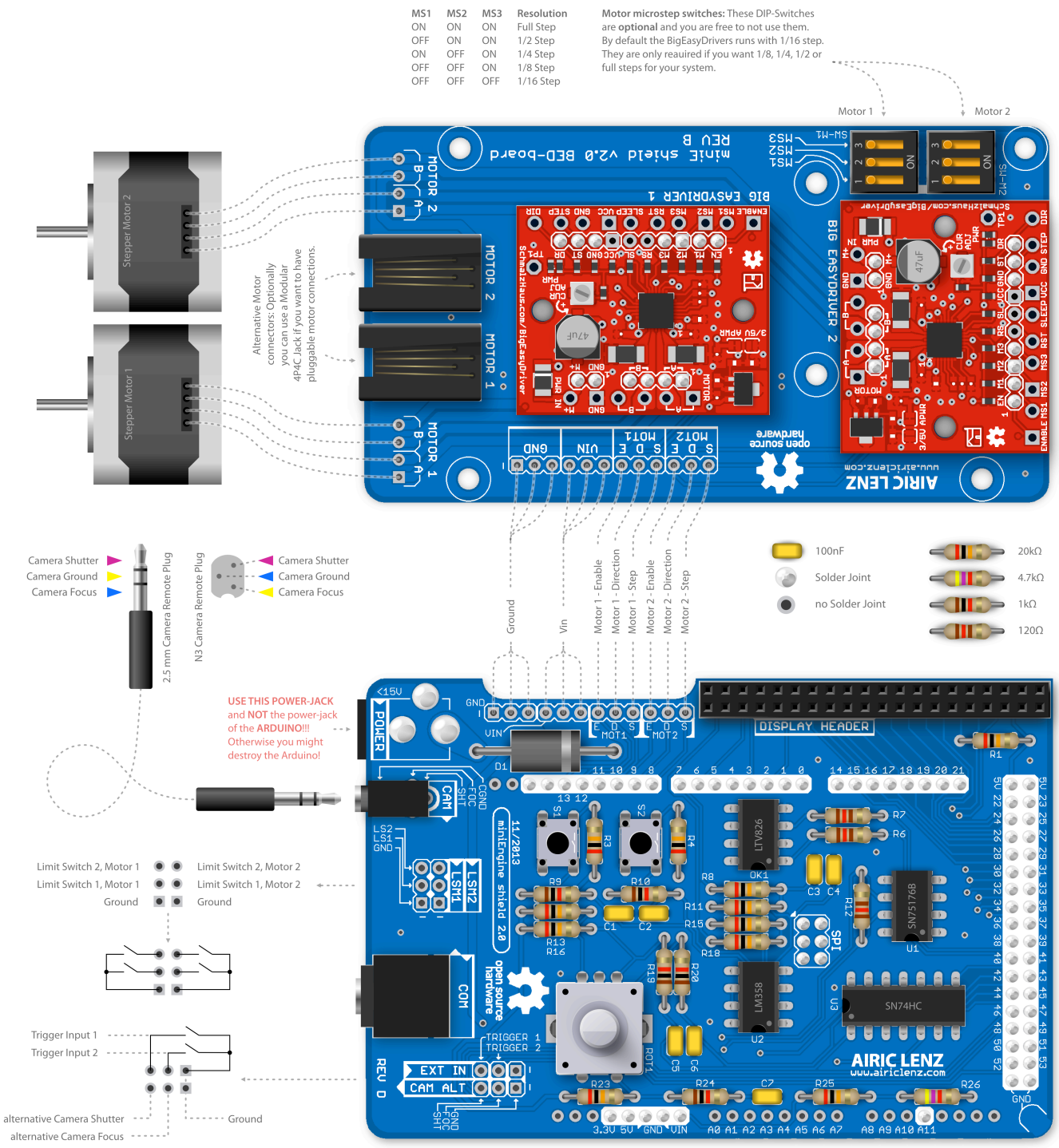
Plug the components together:

Shield: When you have checked that everything was soldered correctly, plug the shield carfully into your Arduino DUE. Be aware that unplugging it again is not easy as many pins are willing to stay connected to the Arduino.

Display: Switch the ITDB02-2.4E display to 16 bit and then plug it carefully to the DISPLAY HEADER of your shield. Be aware that unplugging it again is not easy as it was designed for secure placement in this header.



| 2.3 | *Shield functions*



3 Software

3.1 *Installation*

If you don‘t already have installed the Arduino Development Suite (for the Arduin DUE the beta version 1.5.5 is needed), please download it from

http://arduino.cc

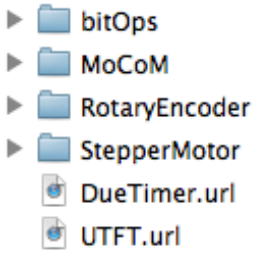
and install it. The latest miniEngine 2 software can be downloaded from here:

http://github.com/airiclenz/miniEngine2

Now, after you downloaded the project, please unzip (if you downloaded a ‚ZIP-archive) or copy it to any location you want. Please open the new folder and then open

*Software \ libraries *

You should then see the following folder content:



4 of the 6 required libraries were developed by me and are already placed in the download but two of them are not. Please open the two files

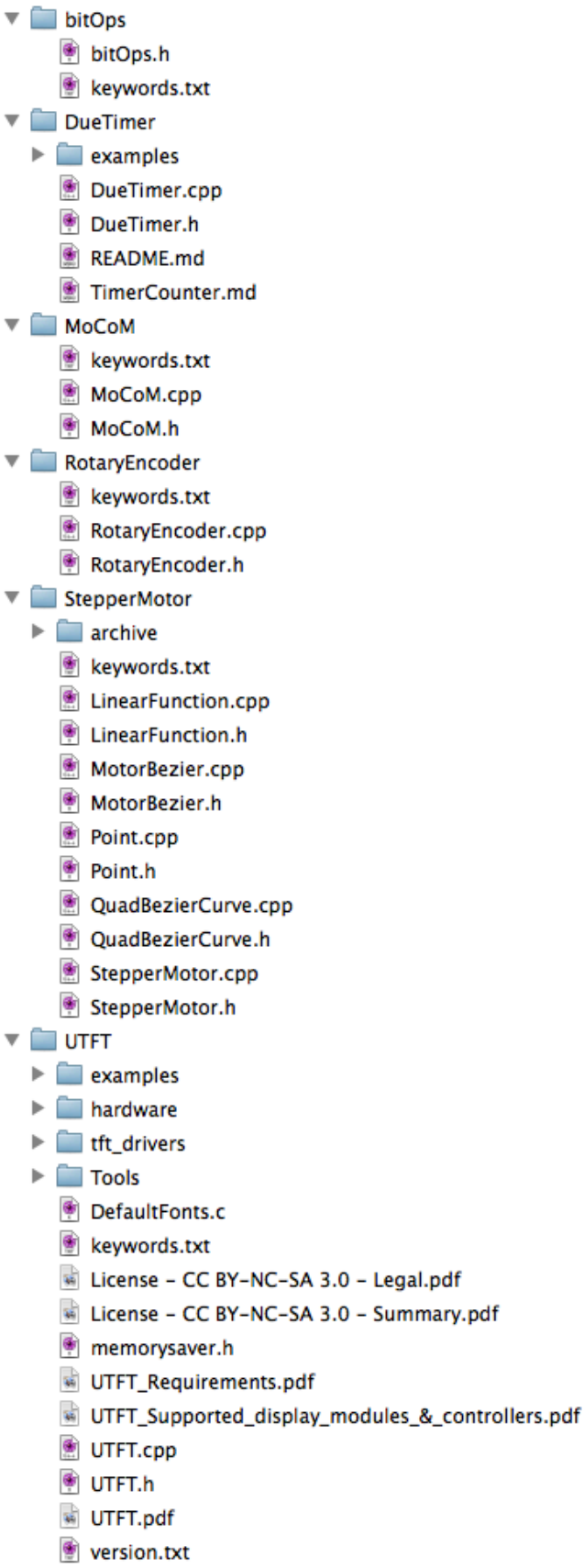
DueTimer.url *https://github.com/ivanseidel/DueTimer*

and

UTFT.url *http://www.henningkarlsen.com/electronics/library.php?id=52*

and download the libraries from there. Unzip them (Rename the Folder „DueTimer-Master“ to „DueTimer“) and place them in the same folder where the other 4 libraries are placed.

You should now have this folder structure with all 6 libraries in your library folder:



To successfully compile the miniEngine 2 code, you need to install these 6 libraries within your Arduino development environment. To do so, open the Arduino folder. This folder should be placed in your user accounts documents folder (You can check the path in the Arduino settings). On a Mac this folder usually can be found here:

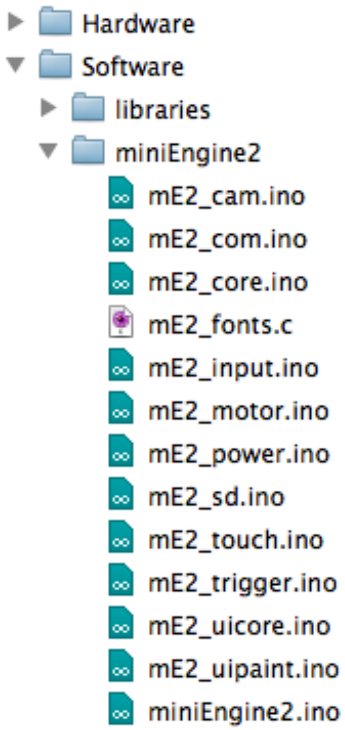
/Users/USERNAME/Documents/Arduino

On a Windows system this folder should be placed here:

Drive:\Users\USERNAME\My Documents\Arduino

In this folder should be a sub folder called *libraries* (if not, create a folder called *libraries* in the Arduino folder). Now copy all 6 library folders from your download folder into this Arduino-library folder. Then re- start the Arduino programming environment.

Now you installed the needed libraries and should be able to compile and install the miniEngine software to your Arduino Due. In the folder you downloaded, go to the sub-folder *Software\miniEngine2*:



Open the project in your Arduino 1.5.5 by double clicking the main project file *miniEngine2.ino* (all needed files will then be loaded automatically).

For uploading the software to your Arduino DUE board, connect your Arduino DUE with the „Programming port (the right one) to your computer (for details about connecting the Arduino DUE, please check the Arduino website and the Arduino documentation). Now press the upload button:

After the code was uploaded successfully, your Arduino board is ready for doing timelapse photography for you.

Software License

The miniEngine software is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

The miniEngine software and all belonging material is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see *http://www.gnu.org/licenses/*.

Hardware License

All hardware designs and schematics provides along with the miniE software are free. You can redistribute it and/or modify it under the terms of the CERN Open Hardware Licence 1.1, or (at your option) any later version.

The hardware documentation and hardware files are distributed in the hope that they will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the CERN Open Hardware Licence for more details.

You should have received a copy of the CERN Open Hardware Licence along with these files. If not, see *http://www.ohwr.org/projects/cernohl/wiki/*.