mini*E*

mini*Engine* - an open-source motion control system for timelapse photography

Documentation of version 1.2

Airic Lenz

Content

| 1 | Introduction

miniE is an engine sketch to provide control over a single stepper motor and camera, using an Arduino board and a DFRobot LCD Keypad shield. It is a minimalist design providing all the necessary features to do shoot-move-shoot or continuous timelapse with a built-in user interface. It was developed to provide a slim and easy way to timelapse photography. The first version of miniE was inspired by the OpenMoCo timelapse engine. For more details please have a look at *www.airiclenz.com*.

| 2 | Needed hardware

For running the miniEngine you will need an Arduino Uno Board and the DFRobot Display shield for a minimal setup. With these components (and some breadboard magic) you are able to use the system as an intervalometer for your camera. To avoid the open wires and the breadboard I suggest the usage of a Arduino proto-shield for placing the electronics.

For a full setup that uses all capabilities of the current version, you will need the following components:

1x Arduino Uno

1x DFRobot Display Shield

1x RTC module (only needed for timed programs)

1x Stepper motor driver (e.g. EasyDriver)

1x Stepper motor fitting your requirements

1x Dolly system to control

Computer running Windows, Linux, or OSX compatible with the Arduino IDE (≥ 1.0)

Some additional material for wiring up all components (e.g. proto Shield, wires, …)

1x Camera remote cable (u can use a cheap remote for you camera model to get one)

You will also need some electric parts for setting up the required connections to all the external components (like camera, motor driver, ...). Here is a list of all parts needed (for more details have a look at the circuit diagram in the APPENDIX of this document):

CAM	2.5 mm headphone jack (DigiKey PartNo.: CP1-2503A-ND)
LIMIT-SWITCHES	2.5 mm headphone jack (DigiKey PartNo.: CP1-2503A-ND)
RTC	Sparkfun RTC MODULE (OPTIONAL)
MOTOR-CONTROL	Terminal screw block 1×3 (pitch 2.54 mm)
MOTOR-ENERGY	Terminal screw block 1×2 (pitch 2.54 mm)
OK1	4N35 Optocoupler (DIL-6)
OK2	4N35 Optocoupler (DIL-6)
R1	470 Ω Resistor (0207/10)
R2	470 Ω Resistor (0207/10)
R3	20 kΩ Resistor (0207/10)
R4	20 kΩ Resistor (0207/10)
R5	20 kΩ Resistor (0207/10; OPTIONAL)
S-DY1	12V 0.2A diode (e.g. fast zener or schottky diode); This part is OPTIONAL and by default bridged on the miniE shield
S-DY2	12V 0.2A diode (e.g. fast zener or schottky diode); This part is OPTIONAL and by default bridged on the miniE shield

If you plan to do timelapse photography from a tripod without any movement of the camera, you will of course not need the movement components like stepper motor, driver board or the dolly system.

| 3 | Installation

If you don‘t already have installed the Arduino Development Suite, please download it from

http://arduino.cc

and install it. The miniE software can be downloaded from the OpenMoCo SourceForge subVersion repository:

http://openmoco.svn.sourceforge.net/viewvc/openmoco/miniE/trunk/miniE

Now, after you downloaded the project, please unzip (if you downloaded a tarBall) or copy it to any location you want. To successfully compile the miniE code, you need an external library, that is used within miniE. To be able to install the external library, please download it from:

https://github.com/jcw/rtclib

After you downloaded the library, you need to install it. To do so, unzip it. It should be in a folder of its own. 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:

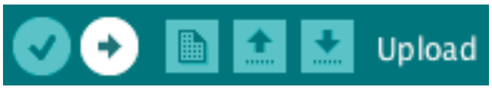
/Users/USERNAME/Documents/Arduino

On a Windows system this folder should be placed here:

Drive:\Users\USERNAME\My Documents\Arduino

If there is already a folder there called *libraries*, place the *RTClib* folder in there. If not, create a folder called *libraries* in the Arduino folder, and drop the *RTClib* folder in there. Then restart the Arduino programming environment.

Open the project by double clicking the main project file *miniE.ino* (all needed files will be loaded automatically then). For uploading the software to your Arduino board, connect your Arduino board (for details about connecting the Arduino, please check the Arduino website and the Arduino documentation) to your computer. Now press the upload button:



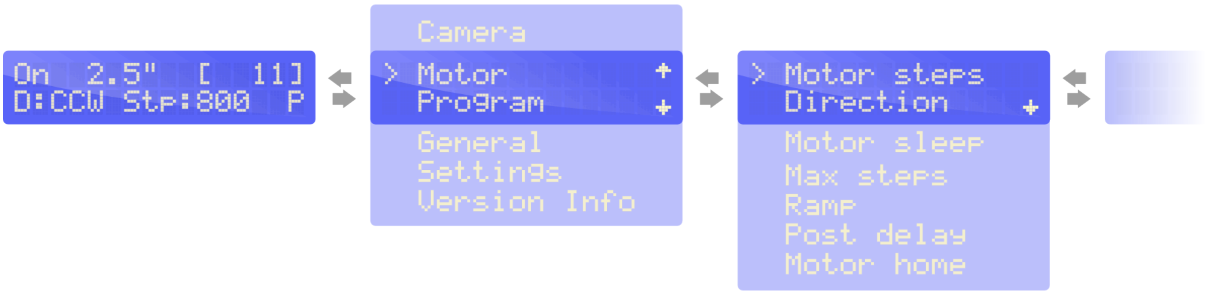
After the code was uploaded successfully, your Arduino board is ready for doing timelapse photography for you.

If you don‘t see anything on your screen, please adjust the LCD contrast.

| 4 | User interface and navigation

The miniE user interface is based on a 2x16 char LCD display and 5 buttons. The full system control is available through this interface. Due to the limited space, available on such a LCD display, the full amount of data and settings is spread over different single setting screens which are available by navigating through a menu tree.

The navigation is based on the general principle that you move sideways through the user interface. A RIGHT key press brings you deeper into the menu structure while a LEFT key press brings you back up. The UP and DOWN keys are used for moving up and down on a menu- or information screen and for changing settings (toggle, increment, decrement). The complete menu tree can be found in Appendix 1 at the end of this document.

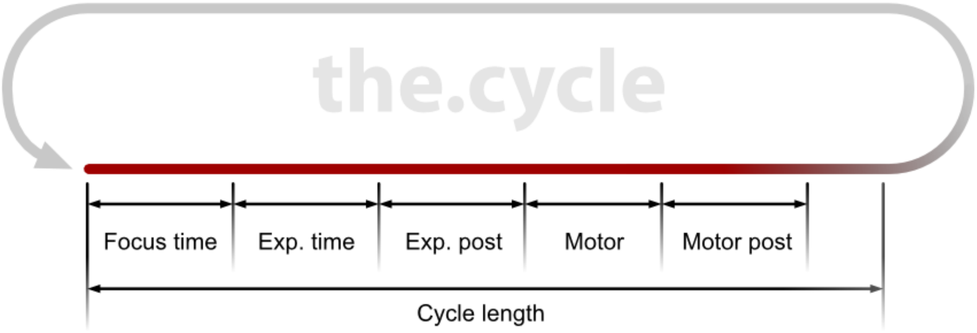


The navigation principle

| 5 | The Cycle

The way miniE works can be described as a cycle. Please keep in mind that the total cycle length, you can define, is not a limiting factor. If "Focus time", "Exposure time", "Exposure post delay", the time the motor needs for the movement and the "Motor post delay" are in total longer than the cycle, all these steps will be performed fully in any case. If this happens, the "real“ cycle length is longer than the time you defined for it and equals the sum of all cycle steps.

The picture below shows the ideal situation where all the single steps take in total less time than the defined cycle length.

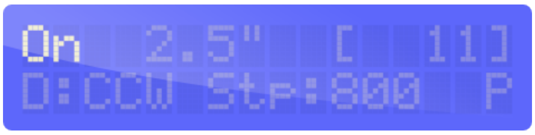


The cycle of miniE

| 6 | Functionalities

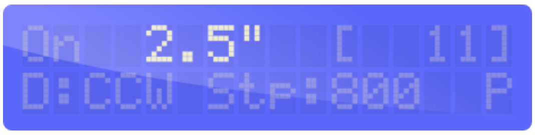
| 6.1 | The main screen

The main screen is the control central of the miniE system. Here you can start and stop the engine, follow the process and view the most important settings and values. You start and stop the program by pressing the SELECT key.



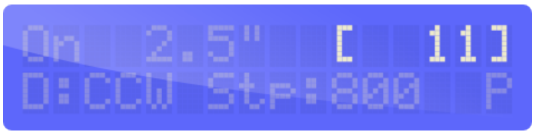
```
On  2.5"  [  11]
D:CCW Stp:800  P
```

Status of the engine. This shows if the engine is running or stopped. You can start / stop it by pressing the SELECT key or through a program



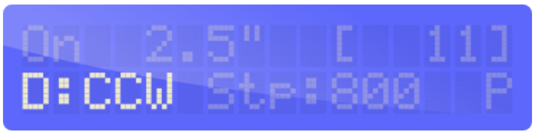
```
On  2.5"  [  11]
D:CCW Stp:800  P
```

Length of the full cycle. You can change this value under [Camera] - [Cycle length]



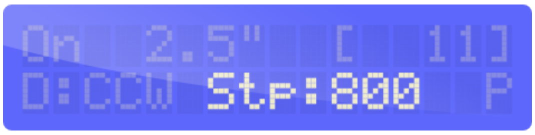
```
On  2.5"  [  11]
D:CCW Stp:800  P
```

Number of shots taken since the engine start. This value will be reset every time the engine re-starts



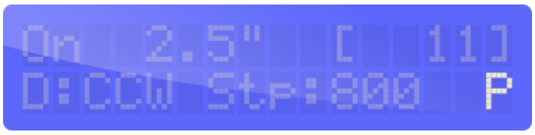
```
On  2.5"  [  11]
D:CCW Stp:800  P
```

Direction of the motor. CW = clockwise; CCW = counter/anti-clockwise; This can be changed under [Motor] - [Direction]



```
On  2.5"  [  11]
D:CCW Stp:800  P
```

The number of steps to do during one cycle. This value can be changed under [Motor] - [Motor steps]



```
On  2.5"  [  11]
D:CCW Stp:800  P
```

Program status. You can only see something here if at least one future program is programmed. If the engine is stopped, you see a static „P“. If you started the engine manually, you will see a blinking „!“ because you possibly prevent a timed program from running. If a timed program is running you will see a blinking „P“

| 6.2 | Camera Functionalities / Settings

Cycle length

The cycle length defines the length of the main loop of the engine. A cycle length of 5 seconds means that the engine will take a photo every 5 seconds. After that all other cycle tasks, like motor movement, will be performed. After that the engine will wait until the 5 seconds are over (counted from the start of the cycle) and start a new cycle then.

Focus time

The focus time is the first task during a cycle. It will trigger the focus line of the camera for the amount of milliseconds you defined. This can be used for focussing or to wake up the camera in very long cycles. If you set the focus time to zero, this task will not be performed.

Exposure time

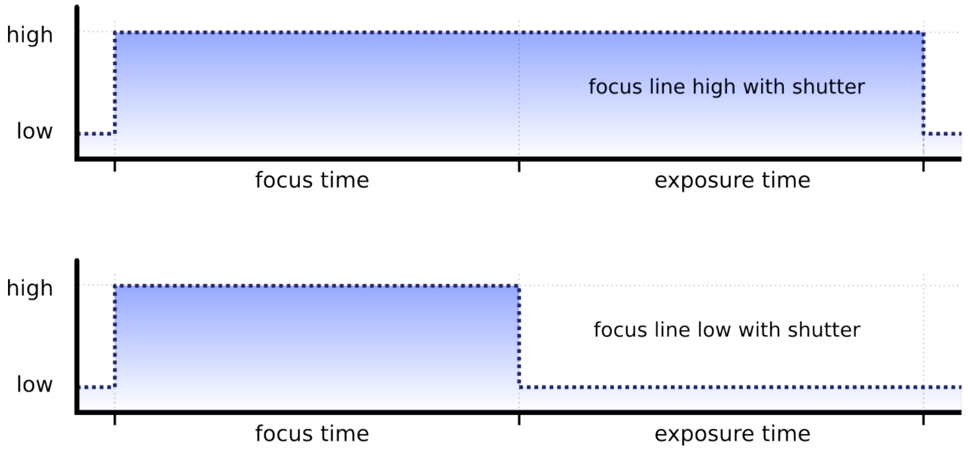
The exposure time is executed directly after focussing. It will trigger the exposure line of the camera for the amount of milliseconds you defined. This is actually triggering the shutter and doing the shot. You can use this to trigger your camera to do a shot in manual mode. In this case you have to set up your camera with your preferred manual settings.

For triggering the shutter, a minimal exposure time of about 40 milliseconds (1/25) is needed. This value changes from camera brand to camera brand.

If you want to do HDR shoots, you have to set up you camera to do all needed shots after one trigger signal. Another solution would be to expose as long as your camera needs to doo all HDR shots (set the camera drive to multi shot mode for this).

Focus-line behaviour

This is setting the behaviour of the focus line during the exposure. You can choose between „high with shutter“ and „low with shutter“. This option is needed due to the fact that different camera brands need different signal combinations to accept the shutter signal.



Focus line behaviour

Max shots

This is defining a stop limit for the engine. After reaching this amount of shots, the engine will be stopped automatically. A value of „0“ means that there is no limit.

Post delay

The post delay is a delay time after the exposure time is over. This can be used to ensure the camera has enough time to finnish the exposure before the motor starts to move.

Test shot

Here you can check if your camera is connected properly and the settings are OK for your specific camera brand. This will do one single exposure with the duration of 1/10th of a second (100 milliseconds).

6.3 | Motor Functionalities / Settings

Motor steps

This setting is for defining the amount of steps to do during your motor phase of the cycle. 1,600 steps are exactly one revelation with most of the stepper motors.

Motor direction

Here you can set the direction of you motor. You can choose between clockwise and counter-clockwise.

Motor post delay

This delay is usually used to let the dolly settle down after motor movement to avoid vibrations.

Motor home menu

This menu contains all settings and functionalities for controlling the motor home behaviour. The motor home position is a free choosable position of the motor. The engine counts every step away from this position while the engine is running (started manually or by a program). You can move the motor back to its home position manually or automatically after a timed program finished.

The motor home position is cleared when miniE is powered up. This means that your motor always starts from home after miniE was powered up.

Move to home

With this functionality you can move the motor back to its home position manually.

Set motor home

This sets the current position of the motor as home position.

Motor sleep

Motor sleep defines the behaviour of the driver board when the motor is not moved. This setting is only valid for the time when the engine is running. While the engine is stopped, the motor is always in sleep mode.

If your motor is in sleep mode between the movements, its energy consumption is drastically reduced.

You can turn sleep mode off to keep the motor powered as long as the engine is running. This is useful if you want to avoid any movements of your dolly during the run.

Max steps

This is defining a stop limit for the engine. After reaching this amount of steps, the engine will be stopped automatically. A value of „0“ means that there is no limit.

Motor ramp

This value is defining the amount of steps your motor will accelerate and decelerate through a motor movement phase.

If you defined a value of 200 steps for ramping and you are doing 1000 steps each cycle, your motor phase looks like this:

- 200 steps acceleration
- 600 steps max speed
- 200 steps deceleration

The motor ramping softly shaped for smooth movements.

Max speed delay

This value defines the time between every motor step at maximum speed. The shorter the delay is, the faster your motor is running.

Min speed delay

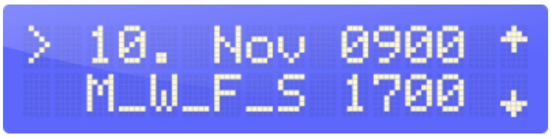
This value defines the time between every motor step at minimum speed (At the very beginning and at the very end of every motor phase). The longer the delay is, the slower your motor is running.

6.4 | Programs

Programs are used to start the engine automatically at a defined time and date. Programs only work if you have a Real-Time-Clock connected to your Arduino. You can define two types of programs:

- ***single date program*** - is only started once
- ***repeating weekday program*** - is started at the defined weekdays (e.g. every Monday)

You can define 30 programs. A programs is not started while you are editing it!



```
> 10. Nov 0900 ↑
  M_W_F_S 1700 ↓
```

In the program list

In the graphic obove you can see the two different program types with their corresponding representation in the program list (24h time format):

- `10. Nov 0900` - *Single date program* - this program will only be started once (at the 10th November at 09:00 o‘clock in the morning, the year is not displayed in the program list)
- `M_W_F_S 1700` - *Repeating weekday program* - this program is started every Monday, Wednesday, Friday and Sunday at 17:00 o‘clock. A program becomes automatically a repeating weekday program if at least one day of the week is activated for the program (see „Program weekdays“)

Add program

You can add a new program with this functionality. After you added a program, you are automatically editing it.

Program start time

You can edit the program start time here. This is the time the engine will started automatically. If the program is a single date program, you will be able to define a date (this is not the case if you are editing a repeating weekday program).

Program weekdays

Here you can set weekdays at which you want the program to be started. If you deactivate all weekdays, the program becomes a single date program and will only be started once.

Program duration

This value defines the duration in minutes the engine will be active after it was started by the program.

Move motor home after program

If enabled, the motor will automatically move back to its home position (see Motor home for more information) after the program was stopped. The motor will be moved back on any event that stops the program (regular end, limit switch, manually stopped by the user).

Program status

If enabled, the program is active and will be considerated by the internal timer. If you disable a program, it has the same effect as if it were not existing.

Delete program

Here you can delete the program you are currently editing.

6.5 General settings

Operation mode

The operation mode is defining the basic way the miniEngine will work. You can choose between two different modes:

- ***shoot-move-shoot mode*** - the shoot-move-shoot mode is doing all steps of the cycle sequentially. This means the camera is triggered, then the motor moves, and then the cycle begins again.

- ***continuous mode*** - in this mode the motor is moving continually (The amount of defined steps per cycle). The camera is triggered while the motor is moving.

System time

You can set the system time here. This functionality is only available if you have a Real-Time-Clock connected to your system.

Backlight delay

This value defines the time in seconds the system is waiting after the last user action, until shutting off the LCD-backlight.

Backlight power

Here you can set the brightness of your LCD-backlight.

Limit switches

You can turn the limit-switches on or off. If you disable your limit switches, a triggered limit-switch will not stop the running engine.

This functionality is included because a not installed limit switch can cause unwanted limit switch events (the Arduino ports, used for the limit switches, are very sensitive and need to be pulled down to ground through a pull-down resistor with a value between 1kΩ and 50kΩ). If there is no pull-down resistor, it is better to turn off the limit switches.

6.6 Settings

Save settings

This functionality is saving all your defined values and settings to the internal EEPROM of the Arduino board. After powering the system all settings and values are automatically loaded from the EEPROM.

Autosave settings

You can save all settings and values automatically. If you enable this functionality the the settings are automatically saved after you changed any setting or value and left the corresponding setting / value screen.

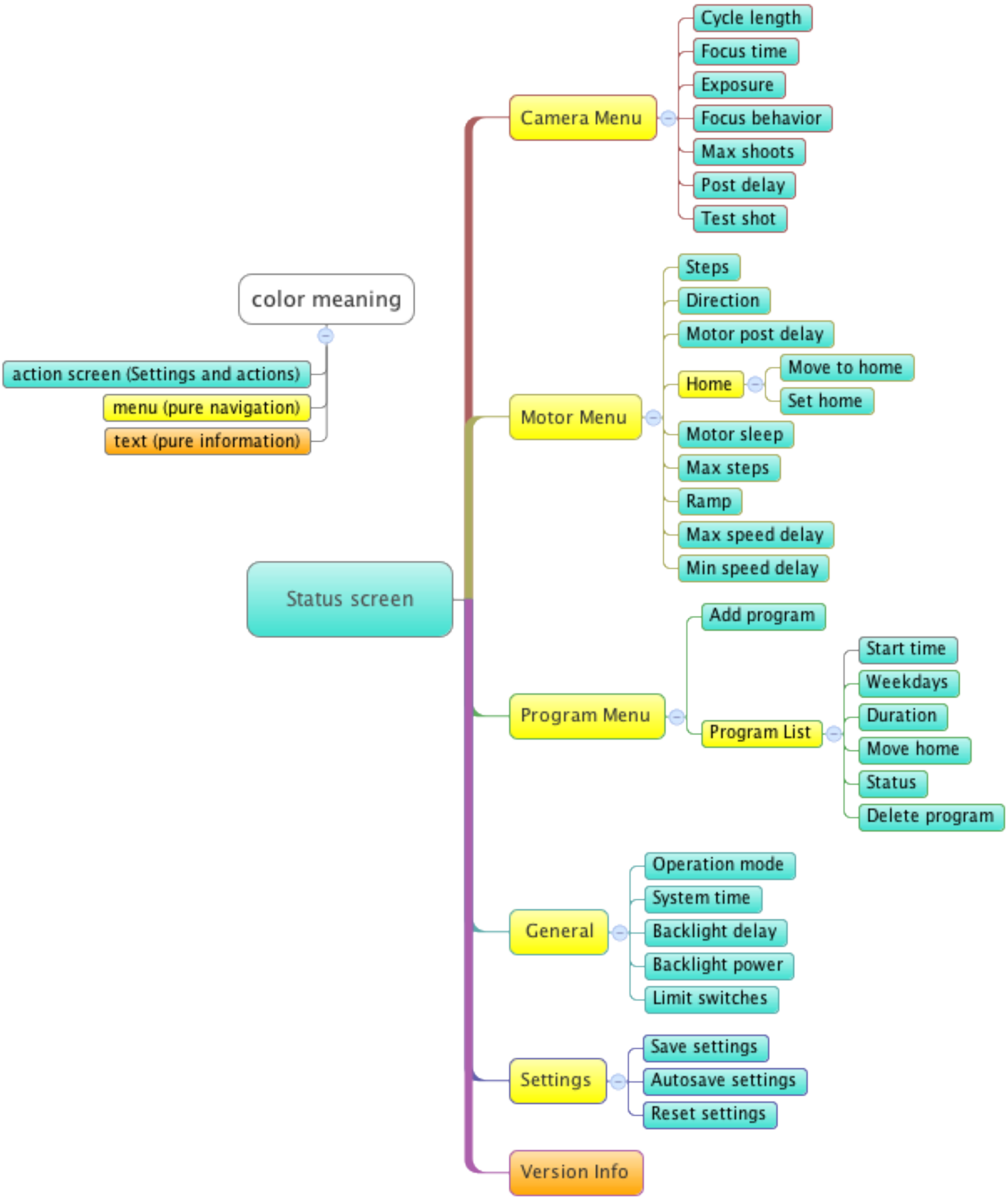
Reset settings

This functionality is resetting all settings and values to the default values. It is also deleting all defined programs.

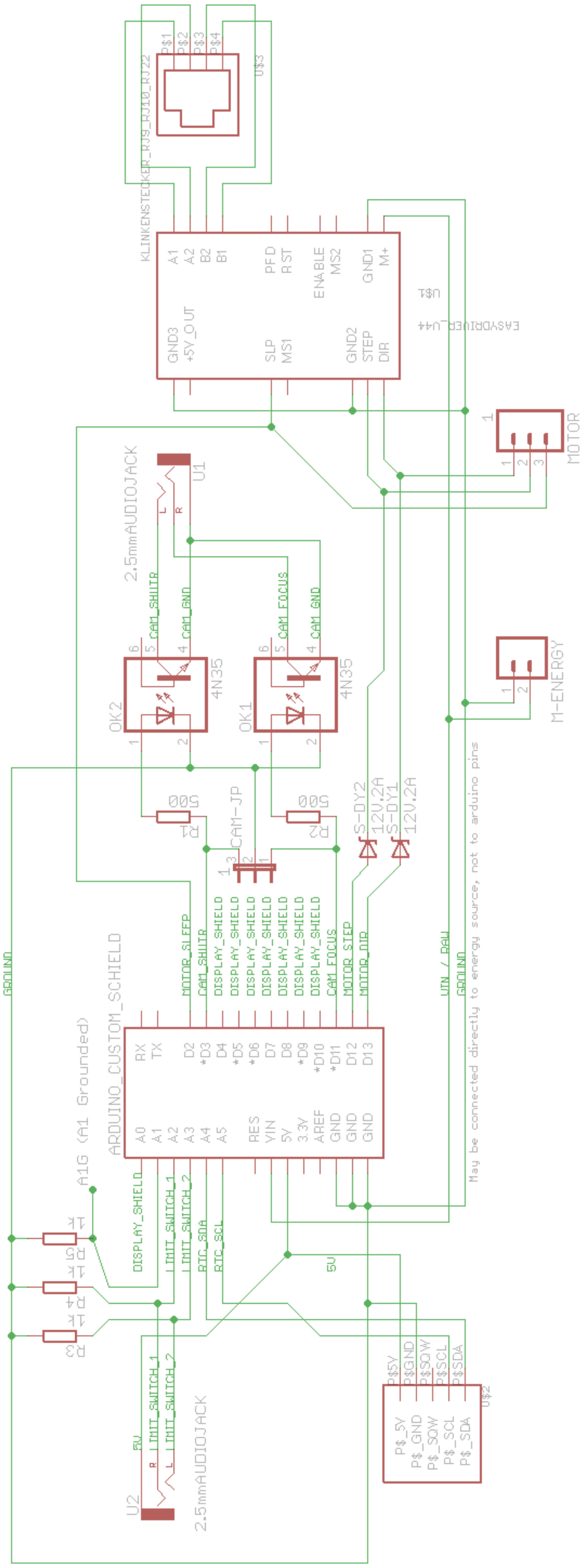
6.7 Version info

Some information about the current installed version of miniEngine.

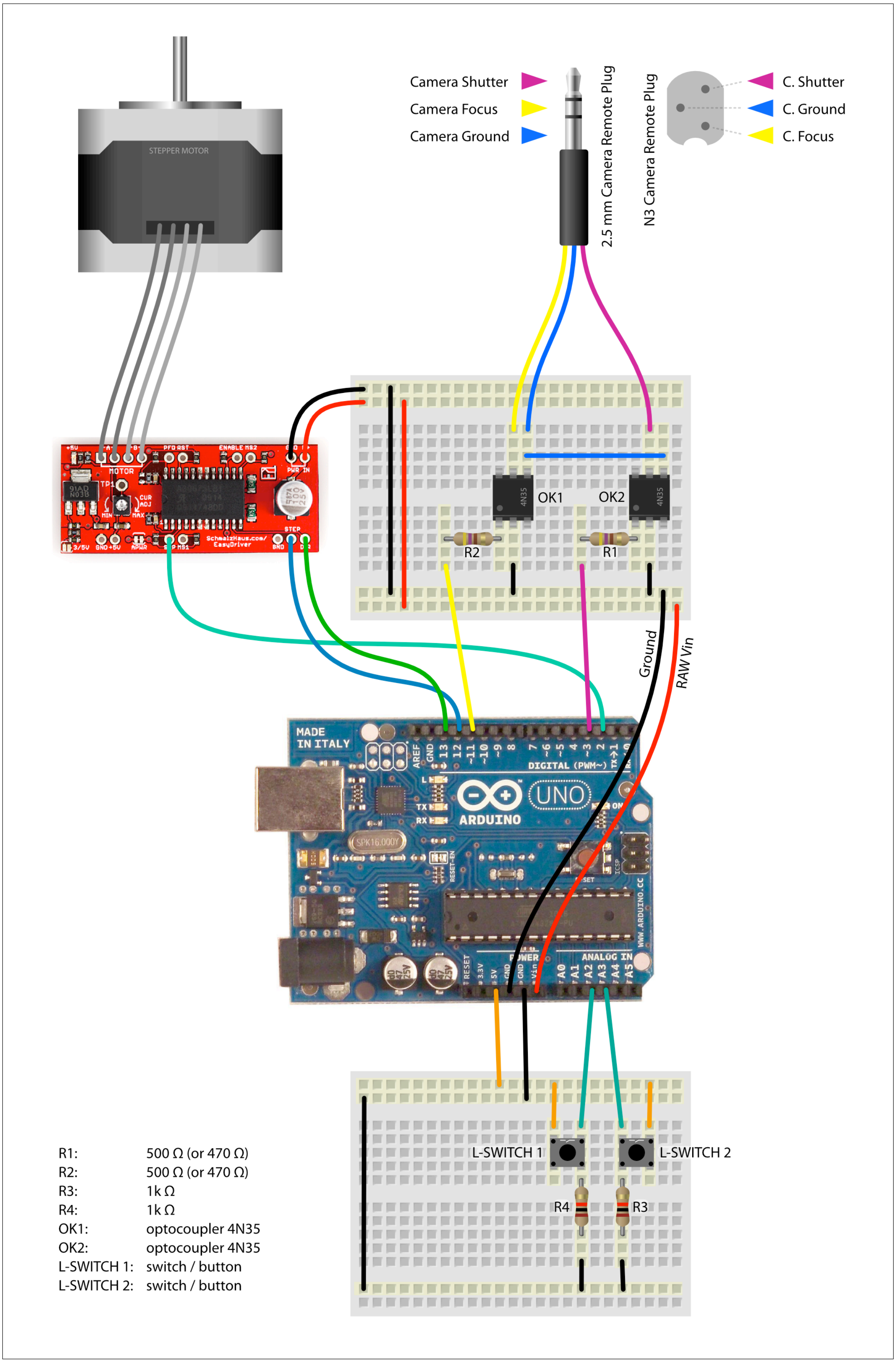
Appendix



APPENDIX 1 - The menu tree



APPENDIX 2 - Circuit diagram



R1:	500 Ω (or 470 Ω)
R2:	500 Ω (or 470 Ω)
R3:	1k Ω
R4:	1k Ω
OK1:	optocoupler 4N35
OK2:	optocoupler 4N35
L-SWITCH 1:	switch / button
L-SWITCH 2:	switch / button

APPENDIX 3 - Arduino breadboard circuit

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/*.