**EdmundOptics16mm Liquid Imaging Lens**

Liquid lens is connected via a flex cable that consists of 4 wires. The middle two are used for the transmitting the voltage generated by the driverboard and the ones on the sides are ground wires. There are two liquids inside the lens – oil and water. Since these liquids are immiscible, they create a very thin interface between themselves. The voltage from the driverboard is applied directly onto this interface, which causes its shape to change rapidly. This property is used for focusing in a matter of tens of miliseconds.

**HV892DB1 – Inductorless, Liquid Lens Driver Demoboard**

The chip has got following input pins:  
– VIN : input voltage, connected to the Raspberry Pi’s 5V output pin

– SCL, SDA – I2C communication pins, connected to respective Raspberry Pi pins

– VDD : reference voltage up to 2.95V, since there is no suitable Raspberry output pin, another 5V pin is used along with a voltage divider and converted into 2.5V  
– GND : connectes to the ground

The two output pins are connected directly onto the liquid lens.  
  
The driverboard processes a single byte information(AMP) send by Raspberry via I2C and generates respective voltage on the output. There is no need to send adress of a register as the driverboard only has one. It is not possible to read any data from the driverboard.

AMP values:

– 0 : stand-by mode, the chip is inactive, there is very little to no output voltage  
– 1 : initiates the chip, the output voltage is ramped up to 9.3V – the outpu voltage at which the liquid lens is at the most convex state possible  
– 2 - 253 : the output voltages between 9.3V and 63V are equally distributed among these values, meaning incrementing the value by one increases the output voltage by approximately 0.211V

– 254 – maximum value which sets the liquid lens to the most concave state possible

Test binary:

The file ‘test\_driverboard/test\_driverboard.c’ contains the source code for building a driverboard testing binary. The program asks what voltage is desired at the output pins and sends such message to the board, that almost exact voltage is applied. There are slight deviations from the average increment of voltage caused by incrementing AMP by one.

**DMM37UX252ML monochrome USB 3.1 camera 3.2MP**

Technical specifications :

resolution : up to 2048x1536px

format : GRAY8/GRAY16\_LE

frames per second : up to 119

Liquid lens is mounted onto the camera using a specific plastic gadget. The camera is connected via USB-C – USB 3.1 cable. If older generations of USB are used, the frames per second will be limited siginficantly. The high framerate is very much needed to utilize the liquid lens to its full potential, as the autofocus algorithm needs to analyze 8-10 frames before successfully focusing.

**Raspberry Pi3**

*Raspberry Pi3 was used instead of its newer generation version because of incompatibility of the BCM2835 library used for I2C communication with the driverboard. If this problem is fixed in the future its preferable to switch to the newer version.*

Raspberry computer is used for powering of the entire optical system, as I2C master device and also for exectution of the auto-focus algorithm. The connection scheme is located at the end of this file.

It will also provide the computational power for the auto-focus algorithm, which will need to be tested to see if it suffices. The camera will be connected to the Raspberry via usb cable. Raspberry will then fetch the data from (liquid lens mounted) camera via available libraries and use the data to analyse the frames and find the optimal focus point.

**Voltage Divider**

The voltage divider is used to divide the 5V output from Raspberry, into 2.5V. The resistors have a resistance of 10 kilo Ohms.

**Software**

Operating system is build via OpenEmbedded(Yocto release Dunfell 3.1). The I2C interface needs to be enabled manually with the build of the system, which is done in the local.conf file.

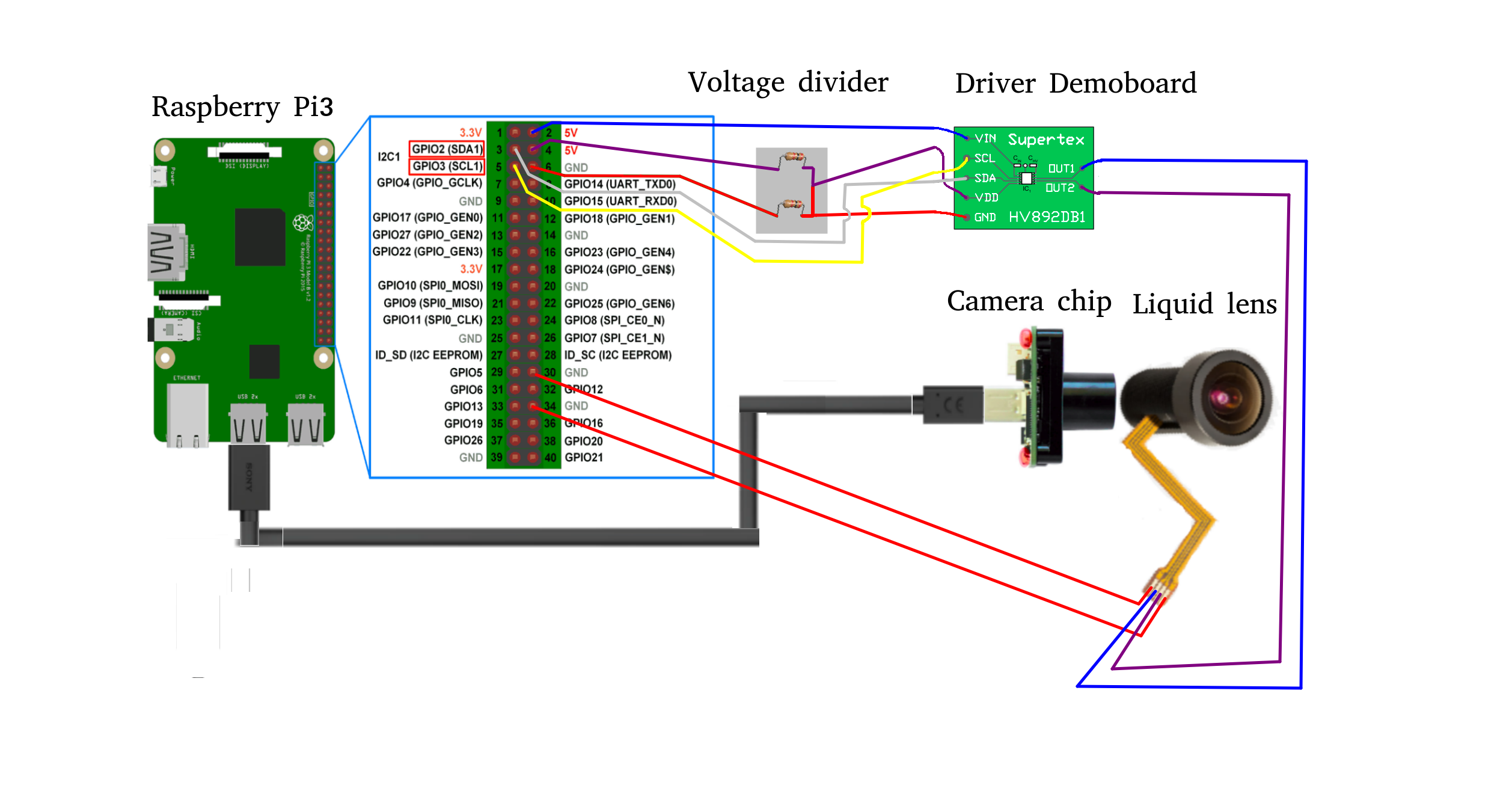
Installed packages: openssh, i2ctools, i2ctools-dev, bcm2835-dev, packagegroup-core-buildessential, python3, python3-pip

outdated recipes – to do: libusb, smbus, tiscamera

**Libraries**

bcm2835, i2cdev

to do: tcamprop, gst

**Detailed assembly**