6 ANNEXES

6.1 Annex I

MEA1K Chip Packaging Protocol

1. Deposit epoxy on the wires to protect them from the liquid: Mix the epoxy with the curing agent on a 10:1 weight ratio (10g+1g for 10 chips). Build a dam around the chip by depositing the epoxy mixture with a syringe. **Do NOT touch the wires with the needle** (will damage the chip and render it useless).

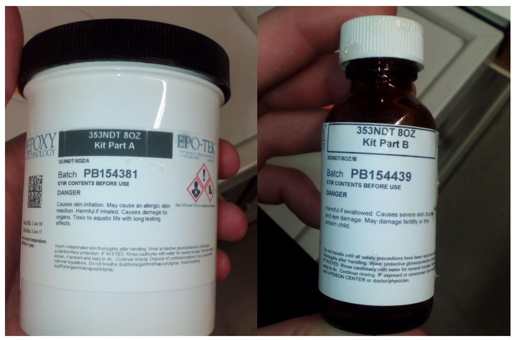

Figure 20: First epoxy (left) and curing agent (right)

Figure 21: 3ml syringe and tip to apply the epoxy mixture

Figure 22: Epoxy dam around the chip

2. Put the chips in the oven for 1 hour at 80ºC.

3. Glue the glass ring around the chip with superglue. Apply the glue across the base of the ring and then align it with the white marks around the chip. **Only one of the edges of the ring is entirely flat**. This is the one that should be glued to the chip.

Figure 23: Chip with glued glass ring

4. Add the second epoxy around the chip. 10:1 epoxy:curing agent ratio again. **The epoxy must cover the wires**. De-gas the epoxy mixture using the vacuum pump before applying it to the chip. The de-gassing will cause a foam to appear on top of the epoxy; open the valve on the pump to prevent the foam from overflowing. When applying the epoxy, **do NOT touch the chip with the syringe**. The level of the epoxy should be $\sim \frac{1}{3}$ of the glass ring. Going above $\frac{1}{2}$ of the glass ring will overflow the first epoxy. Use ~30g of the epoxy mixture for 10 chips.

Figure 24: Second epoxy and curing agent

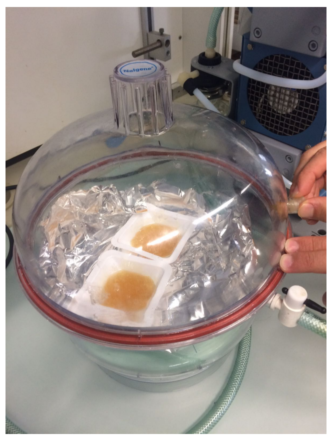

Figure 25: De-gassing the second epoxy mixture

5. Cure at 80ºC for 1 hour, starting with a cold oven if possible to allow the epoxy to settle better. Then cure for 1 hour at 140ºC.
6. Clean the chip with plasma oxygen before plating the cells:
    (a) Place the samples inside the plasma generator
    (b) Switch on the vacuum pump
    (c) Close the air valve
    (d) Wait for 2 minutes
    (e) Switch on the plasma generator
    (f) Wait for 10 minutes
    (g) Switch off the plasma generator
    (h) Slightly open the valve and immediately switch off the vacuum pump

(i) Switch off the vacuum pump, wait for pressure to equilibrate and retrieve the samples

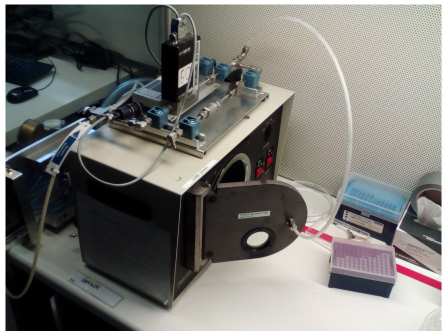

Figure 26: Plasma generator.

7. Add PBS to the chips and test them.

   (a) Add enough PBS to cover the electrodes but do not spill it on the rest of the chip.

   (b) Plug the chip to a setup (**make sure that the connectors are dry**), download a configuration and see that the signals show only white noise and that no strong oscillations are present.

   (c) Clean the PBS from the chips using distilled water to prevent the salts in the PBS from crystallizing on top of the electrodes.

8. Add Platinum black to the working chips:

   - **The platinum solution is toxic, always wear gloves when working in the Pt-black hood!**

   (a) Plug the chip into the setup

   (b) Add 2ml of the platinum solution to the chip

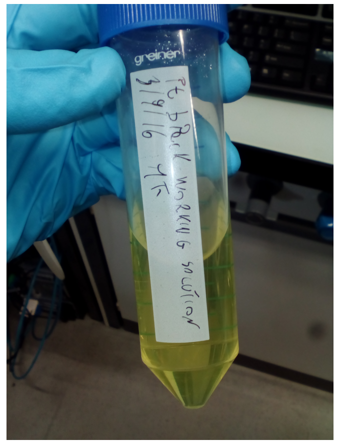

Figure 27: Pt-black solution

(a) Place the electrode from the multi-meter on top of the chip. Make sure the electrode is covered by the platinum solution and **do NOT touch the electrodes with it**

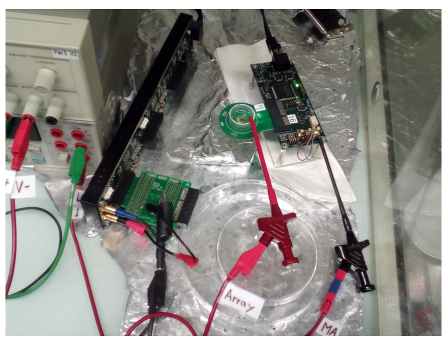

Figure 28: Chip connected to the Pt-black setup. Note the electrode from the multi-meter on top of the chip.

(a) Switch on the FPGA, start the MEA1k server and the pt_black_gui.py script

(b) **Very** gently, clean the surface of the chip with a Q-tip

(c) Check that the voltage on the power supply is set to 1V. Set the multi-meter to current by pressing SHIFT and the "DC V / DC I" key. Press Run on the pt_black_gui script to start running current through the chip and deposit the platinum. Check that the current is around 400 mA

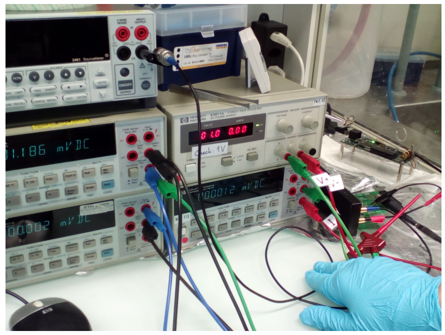

Figure 29: Power supply (right side, top) and multi-meter (right side, bottom)/

(a) Let the platinum deposition take place for around 1 minute. The process might cause bubbles to form on the chip's surface preventing the platinum from reaching it. To prevent this, gently pipet a bit of the solution in and out during the process

(b) Pass the Pt solution to the next chip (it can be reused up to 20 times). Clean the chip thoroughly with distilled water to **remove ALL of the Pt solution** (it is toxic and would damage the cells).

(c) Check the the results under the microscope. The electrodes should be black if properly covered in platinum.

Figure 30: Leica microscope to check the Pt-deposition

(a) Collect the left over Pt solution into the falcon tube labeled as waste, **do NOT throw Pt solution on the sink**

9. Test the chips on the recording setup with PBS. Plug the chip to a setup (**make sure that the connectors are dry**), download a configuration and see that the signals show only white noise and that no strong oscillations are present.

6.2 Annex II

Python script for closed-loop stimulation

```
## Import libraries
# C++ libraries:
import libarray
import mea1k
import ringbuffer
# Import python libraries
import os
import sys
import time
import datetime
import ConfigParser
import mea1kusr.offset
import mea1kusr.save
import mea1kusr.setup
import mea1kusr.electrode
import mea1kusr.init
import mea1kusr.stimulation as stim
import libarray
import npgen
import ringbuffer
import datetime
## Setup the protocol
# Ouput file, electrodes, peak detection
HEXFile   = 'el_selection.hex.nrk'
outputDir = 'closedLoop'
# Make sure that the selected electrodes are routed!
DET_ELECTRODE = 7606 # Electrode to detect
STIM_ELECTRODE = 19908 # Electrode to stimulate
# Delay inside the 'wait' loop
delay_ms = 2000
# Peak detection threshold in XXX units
detectionTh = -40000
# Delay for CloseLoop stimulation (in samples)
close_loop_delay = 2 * 20
# Closed-loop duration in XXX units (seconds?)
cl_length = 20000 * 60 *60
# Stimulation
phase=4  # Not below 50 ms!
ampBits=20 # 1 bit 2.92mV (3/1024)...~60mV
dacOffset=512
## Connect the stimulation electrode
mea1kusr.init.board();
```

```
mea1kusr.init.chip();
# Enable stimulation buffer
mea1k.go( mea1k.cmdCore(
          onChipBias=1,
          stimPowerDown=0,
          outputEn=1,
          spi=0,                      # 0 == DataxDO off
          tx=0,                       # 0 == DAC0
          rstMode=0,                  # 0 == auto
          offsetCyc=7,
          resetCyc=7,
          wlCyc=7))
mea1k.go( mea1k.cmdReadout(
          s1Bypass=0,
          s1Gain=1,                   # 1 == x16
          s1RstMode=0,                # 0 == disconnect
          s2Bypass=0,
          s2Gain=5,                   # 5 == x16
          s3Bypass=0,
          s3Gain=0))                   # 0 ==x2
## Define the stimulation to be triggered in the closed-loop
# Loop contains stimulation (just pulse)
lc   = mea1k.Loop()
lc.clear()
lc.reset()
lc.setStart()  # Indicate start of the loop
lc.toLoop(( mea1k.cmdDelaySamples( 100 ) ))
lc.toLoop(( mea1k.cmdStatusOut( 1 ) ) )
# voltage: baseline to positive value (sent value is of opposite sign)
lc.toLoop( mea1k.cmdDAC( 0, dacOffset-ampBits, 512) )
# pause while deflected
lc.toLoop(( mea1k.cmdDelaySamples( phase ) ) )
# voltage: from positive value negative value
lc.toLoop( mea1k.cmdDAC( 0, dacOffset+ampBits, 512) )
# pause while at negative value
lc.toLoop(( mea1k.cmdDelaySamples( phase ) ) )
# voltage: return from negative to baseline
lc.toLoop( mea1k.cmdDAC( 0 , dacOffset , 512 ) )   # send commands to the loop queue
lc.setStop()   # Indicate end of the loop
lc.download()  # Program the loop into the FPGA
## Load chip configuration and send stimulation config
ring = ringbuffer.ringBuffer()
ring.start()
chip = libarray.Chip()
chip.loadHEX( HEXFile )
stimEl = STIM_ELECTRODE
```

```
chip.electrodeToStim(stimEl)
stimCh = chip.stimNoAtElectrode( stimEl )
readCh = chip.readoutAtStim( stimCh )
chip.download()
stim.offAll().send()            # Switch off all stim channels
stim.onOne( stimCh , 1 ).send() # Turn on only one channel (stimCh)
## Enable Filtering, detection, and closed loop
detEl = DET_ELECTRODE
mea1k.go( mea1k.cmdEnable( 0 ) )
detection_channel = chip.readoutAtElectrode(detEl)
mea1k.go( mea1k.cmdChannel( detection_channel ) )
mea1k.go( mea1k.cmdThresholdLow( -4000 ) )
mea1k.go( mea1k.cmdRefractPeriod( 100 ) ) # 5 ms
mea1k.go( mea1k.cmdDelay( close_loop_delay ) )
mea1k.go( mea1k.cmdEnable( 1 ) )
#########################################################################
# Run until here, to find RMS of readout channel after digital filtering
#########################################################################
## Set RMS-based or arbitrary threshold and re-start the closed-loop stimulation
new_threshold = -200
mea1k.go( mea1k.cmdEnable( 0 ) )
mea1k.go( mea1k.cmdThresholdLow( new_threshold ) )
mea1k.go( mea1k.cmdEnable( 1 ) )
## Start recording
save = mea1kusr.save.save();
save.mkDir( outputDir )
save.reset( )           # reset file counter
save.start(”)
## Wait for the protocol to finish
time.sleep( 1 )
time_zero = time.time()
while (time.time() < (time_zero + cl_length/1000.0) ):
    sys.stderr.write(".")
    time.sleep( (delay_ms) * 1e-3 )
sys.stderr.write("End @ {0}\n".format(ring.getFrameNo() ) )
time.sleep( 4 )
sys.stderr.write("Done with single electrode at: " + str(datetime.datetime.now()))
elapsed = time.time() - time_zero
sys.stderr.write(elapsed)
## Disable close loop and stop recording
mea1k.cmdEnable(0)
save.stop()
stim.offAll().send()
```