Electronic Supplementary Material (ESI) for Analytical Methods.


**SUPPLEMENTARY INFORMATION**

# Fully automated station for testing, characterizing and modifying screen-printed electrodes

Jan Zitka [a,b], Jan Sileny [a], Jiri Kudr [a], Zuzana Koudelkova [a], Lada Ilieva [a], Lukas Richtera [a,b], Tomas Syrovy [c], Vojtech Adam [a,b] and Ondrej Zitka [a,b,*]

[a] Department of Chemistry and Biochemistry, Mendel University in Brno, Zemedelska 1, Brno CZ-613 00, Czech Republic

[b] Central European Institute of Technology, Brno University of Technology, Technicka 3058/10, Brno CZ-616 00, Czech Republic

[c] Faculty of Chemical Technology, University of Pardubice, Studentska 95, Pardubice CZ-532 10, Czech Republic

* e-mail: ondrej.zitka@mendelu.cz

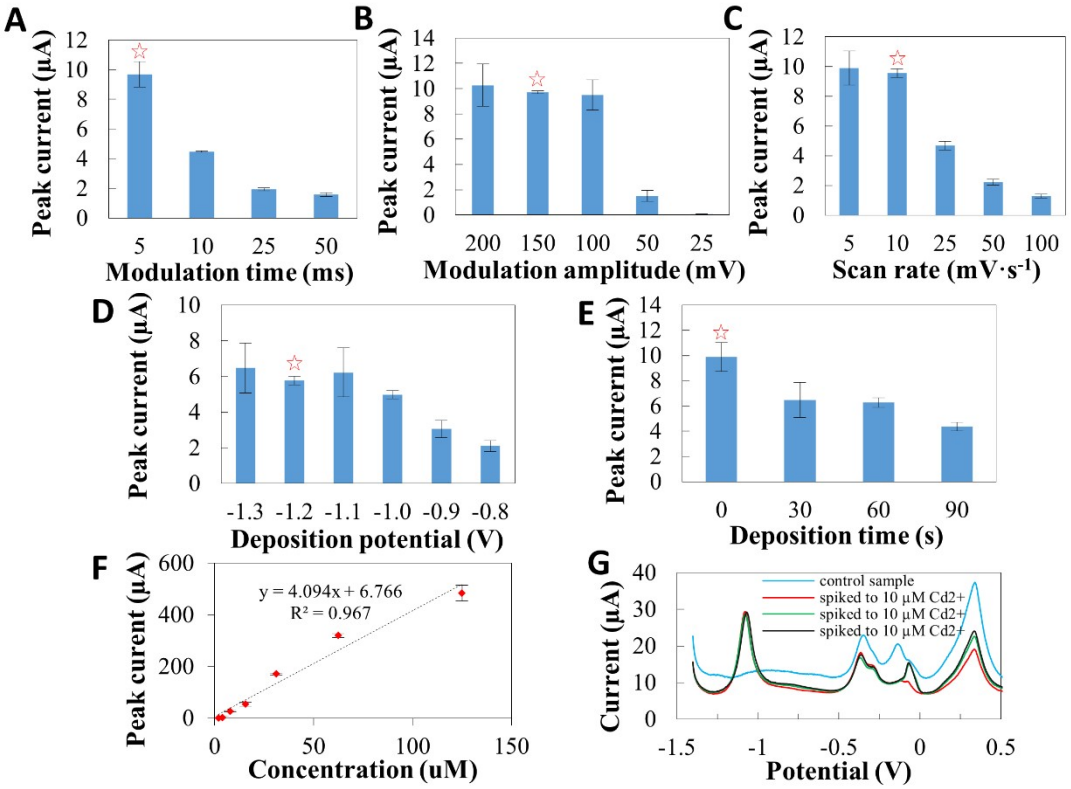


Fig. S1. Optimization of $Cd^{2+}$ analysis using DPV laboratory-made electrodes. (A) Dependence of 5 µM $Cd^{2+}$ peak heights on modulation time (5 – 50 ms) (scan from −1.4 V to 0.5 V, modulation amplitude 100 mV, step potential 2.44 mV, scan rate 10 mV · $s^{-1}$). (B) Dependence of 5 µM $Cd^{2+}$ peak heights on modulation amplitude (25 – 200 ms) (scan from −1.4 V to 0.5 V, modulation time 5 ms, step potential 2.44 mV, scan rate 10 mV · $s^{-1}$). (C) Dependence of 5 µM $Cd^{2+}$ peak heights on scan rate (5 – 100 mV · $s^{-1}$) (scan from −1.4 V to 0.5 V, modulation time 5 ms, modulation amplitude 150 mV, step potential 2.44 mV). (D) Dependence of 5 µM $Cd^{2+}$ peak heights on deposition potential (−1.3 – −0.8 V) (scan from −1.4 V to 0.5 V, deposition time 30 s, modulation time 5 ms, modulation amplitude 150 mV, scan rate 5 mV · $s^{-1}$, step potential 2.44 mV). (E) Dependence of 5 µM $Cd^{2+}$ peak heights on deposition time (0 – 90 s) (scan from −1.4 V to 0.5 V, deposition potential −1.2 V, modulation time 5 ms, modulation amplitude 150 mV, scan rate 5 mV · $s^{-1}$, step potential 2.44 mV). (F) Dependence of $Cd^{2+}$ peak heights on $Cd^{2+}$ concentration (2 – 125 µM) (scan from −1.4 V to 0.5 V, no depostion, modulation time 5 ms, modulation amplitude 150 mV, scan rate 5 mV · $s^{-1}$, step potential 2.44 mV). (G) DPV voltammograms of control sample (0.2 M acetate buffer pH 5 mixed with tap water, 1:1) and same sample spiked with $Cd^{2+}$ (final concentration 10 µM, analyzed in triplicate). The parameters selected as an optimal are marked with star.

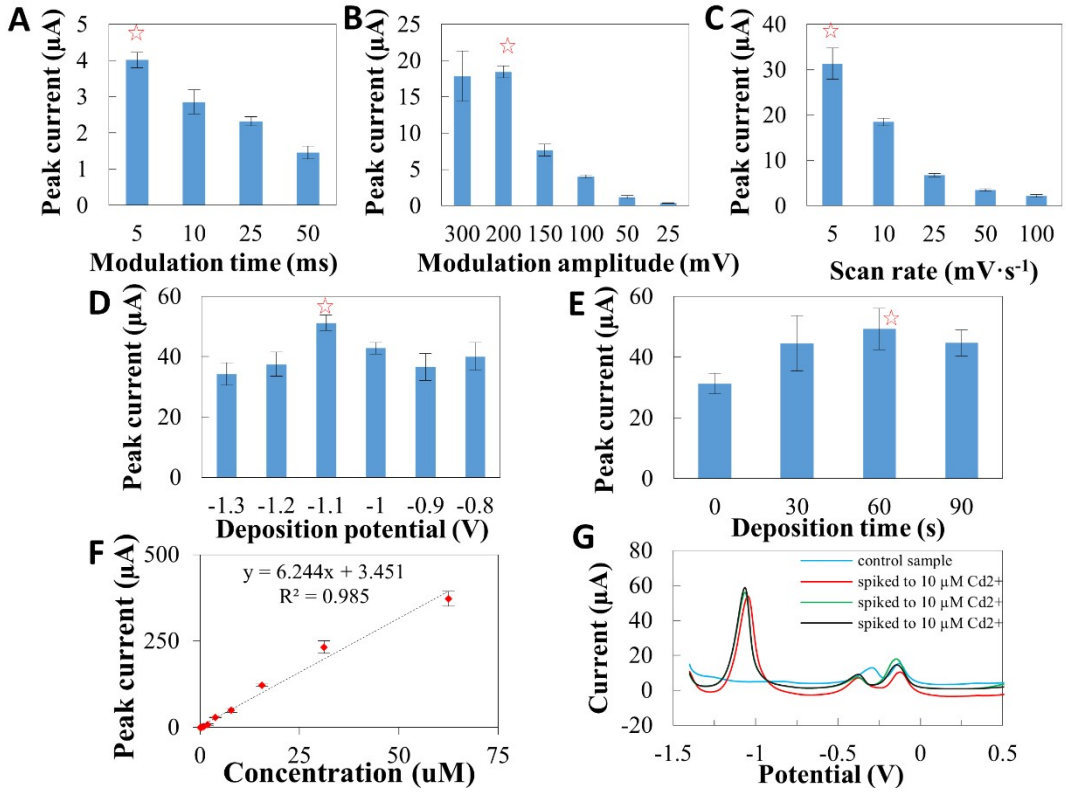


Fig. S2. Optimization of $Cd^{2+}$ analysis using Micrux electrodes. (A) Dependence of 5 µM $Cd^{2+}$ peak heights on modulation time (5 – 50 ms) (scan from −1.4 V to 0.5 V, modulation amplitude 100 mV, step potential 2.44 mV, scan rate 10 mV · $s^{-1}$). (B) Dependence of 5 µM $Cd^{2+}$ peak heights on modulation amplitude (25 – 300 ms) (scan from −1.4 V to 0.5 V, modulation time 5 ms, step potential 2.44 mV, scan rate 10 mV · $s^{-1}$). (C) Dependence of 5 µM $Cd^{2+}$ peak heights on scan rate (5 – 100 mV · $s^{-1}$) (scan from −1.4 V to 0.5 V, modulation time 5 ms, modulation amplitude 200 mV, step potential 2.44 mV). (D) Dependence of 5 µM $Cd^{2+}$ peak heights on deposition potential (−1.3 – −0.8 V) (scan from −1.4 V to 0.5 V, deposition time 30 s, modulation time 5 ms, modulation amplitude 200 mV, scan rate 5 mV · $s^{-1}$, step potential 2.44 mV). (E) Dependence of 5 µM $Cd^{2+}$ peak heights on deposition time (0 – 90 s) (scan from −1.4 V to 0.5 V, deposition potential −1.1 V, modulation time 5 ms, modulation amplitude 200 mV, scan rate 5 mV · $s^{-1}$, step potential 2.44 mV). (F) Dependence of $Cd^{2+}$ peak heights on $Cd^{2+}$ concentration (0.1 – 62.5 µM) (scan from −1.4 V to 0.5 V, deposition potential -1.1 V, deposition time 60 s, modulation time 5 ms, modulation amplitude 200 mV, scan rate 5 mV · $s^{-1}$, step potential 2.44 mV). (G) DPV voltammograms of control sample (0.2 M acetate buffer pH 5 mixed with tap water, 1:1) and same sample spiked with $Cd^{2+}$ (final concentration 10 µM, analyzed in triplicate). The parameters selected as an optimal are marked with star.

Table S1.: Analytical parameters of electrochemical detection of $Cd^{2+}$.

| ***Electrode*** | ***Regression equation*** | ***Linear dynamic range (µM)*** | ***$R^2$ a)*** | ***LOD b)*** | ***LOQ c*** |
|---|---|---|---|---|---|
| laboratory-made | y = 4.094x + 6.766 | 125.0 – 2.0 | 0.967 | 0.4 µM | 1.2 µM |
| Micrux | y = 6.244x + 3.451 | 62.5 – 0.1 | 0.985 | 8.0 nM | 26.9 nM |

a) Regression coefficient.

b) LOD (S/N=3).

c) LOQ (S/N=10)

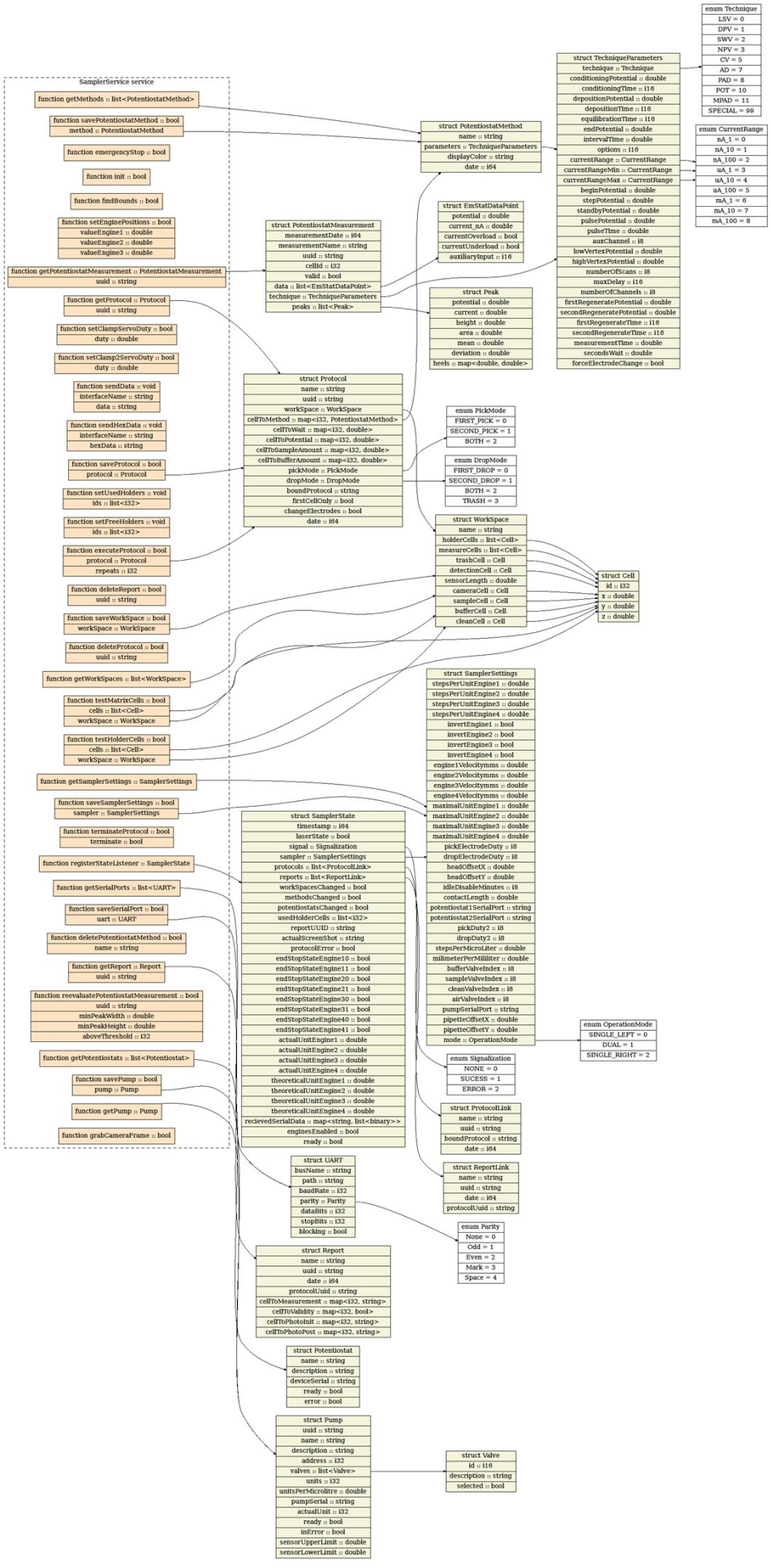
SamplerService service
function getMethods :: list<PotentiostatMethod>
function savePotentiostatMethod :: bool
method :: PotentiostatMethod
function emergencyStop :: bool
function init :: bool
function findBounds :: bool
function setEnginePositions :: bool
valueEngine1 :: double
valueEngine2 :: double
valueEngine3 :: double
function getPotentiostatMeasurement :: PotentiostatMeasurement
uuid :: string
function getProtocol :: Protocol
uuid :: string
function setClampServoDuty :: bool
duty :: double
function setClamp2ServoDuty :: bool
duty :: double
function sendData :: void
interfaceName :: string
data :: string
function sendHexData :: void
interfaceName :: string
hexData :: string
function saveProtocol :: bool
protocol :: Protocol
function setUsedHolders :: void
ids :: list<i32>
function setFreeHolders :: void
ids :: list<i32>
function executeProtocol :: bool
protocol :: Protocol
repeats :: i32
function deleteReport :: bool
uuid :: string
function saveWorkSpace :: bool
workSpace :: WorkSpace
function deleteProtocol :: bool
uuid :: string
function getWorkSpaces :: list<WorkSpace>
function testMatrixCells :: bool
cells :: list<Cell>
workSpace :: WorkSpace
function testHolderCells :: bool
cells :: list<Cell>
workSpace :: WorkSpace
function getSamplerSettings :: SamplerSettings
function saveSamplerSettings :: bool
sampler :: SamplerSettings
function terminateProtocol :: bool
terminate :: bool
function registerStateListener :: SamplerState
function getSerialPorts :: list<UART>
function saveSerialPort :: bool
uart :: UART
function deletePotentiostatMethod :: bool
name :: string
function getReport :: Report
uuid :: string
function reevaluatePotentiostatMeasurement :: bool
uuid :: string
minPeakWidth :: double
minPeakHeight :: double
aboveThreshold :: i32
function getPotentiostats :: list<Potentiostat>
function savePump :: bool
pump :: Pump
function getPump :: Pump
function grabCameraFrame :: bool
struct PotentiostatMethod
name :: string
parameters :: TechniqueParameters
displayColor :: string
date :: i64
struct TechniqueParameters
technique :: Technique
conditioningPotential :: double
conditioningTime :: i16
depositionPotential :: double
depositionTime :: i16
equilibrationTime :: i16
endPotential :: double
intervalTime :: double
options :: i16
currentRange :: CurrentRange
currentRangeMin :: CurrentRange
currentRangeMax :: CurrentRange
beginPotential :: double
stepPotential :: double
standbyPotential :: double
pulsePotential :: double
pulseTime :: double
auxChannel :: i8
lowVertexPotential :: double
highVertexPotential :: double
numberOfScans :: i8
muxDelay :: i16
numberOfChannels :: i8
firstRegeneratePotential :: double
secondRegeneratePotential :: double
firstRegenerateTime :: i16
secondRegenerateTime :: i16
measurementTime :: double
secondsWait :: double
forceElectrodeChange :: bool
enum Technique
LSV = 0
DPV = 1
SWV = 2
NPV = 3
CV = 5
AD = 7
PAD = 8
POT = 10
MPAD = 11
SPECIAL = 99
enum CurrentRange
nA_1 = 0
nA_10 = 1
nA_100 = 2
uA_1 = 3
uA_10 = 4
uA_100 = 5
mA_1 = 6
mA_10 = 7
mA_100 = 8
struct PotentiostatMeasurement
measurementDate :: i64
measurementName :: string
uuid :: string
cellId :: i32
valid :: bool
data :: list<EmStatDataPoint>
technique :: TechniqueParameters
peaks :: list<Peak>
struct EmStatDataPoint
potential :: double
current_nA :: double
currentOverload :: bool
currentUnderload :: bool
auxiliaryInput :: i16
struct Peak
potential :: double
current :: double
height :: double
area :: double
mean :: double
deviation :: double
heels :: map<double, double>
struct Protocol
name :: string
uuid :: string
workSpace :: WorkSpace
cellToMethod :: map<i32, PotentiostatMethod>
cellToWait :: map<i32, double>
cellToPotential :: map<i32, double>
cellToSampleAmount :: map<i32, double>
cellToBufferAmount :: map<i32, double>
pickMode :: PickMode
dropMode :: DropMode
boundProtocol :: string
firstCellOnly :: bool
changeElectrodes :: bool
date :: i64
enum PickMode
FIRST_PICK = 0
SECOND_PICK = 1
BOTH = 2
enum DropMode
FIRST_DROP = 0
SECOND_DROP = 1
BOTH = 2
TRASH = 3
struct WorkSpace
name :: string
holderCells :: list<Cell>
measureCells :: list<Cell>
trashCell :: Cell
detectionCell :: Cell
sensorLength :: double
cameraCell :: Cell
sampleCell :: Cell
bufferCell :: Cell
cleanCell :: Cell
struct Cell
id :: i32
x :: double
y :: double
z :: double
struct SamplerSettings
stepsPerUnitEngine1 :: double
stepsPerUnitEngine2 :: double
stepsPerUnitEngine3 :: double
stepsPerUnitEngine4 :: double
invertEngine1 :: bool
invertEngine2 :: bool
invertEngine3 :: bool
invertEngine4 :: bool
engine1Velocitymms :: double
engine2Velocitymms :: double
engine3Velocitymms :: double
engine4Velocitymms :: double
maximalUnitEngine1 :: double
maximalUnitEngine2 :: double
maximalUnitEngine3 :: double
maximalUnitEngine4 :: double
pickElectrodeDuty :: i8
dropElectrodeDuty :: i8
headOffsetX :: double
headOffsetY :: double
idleDisableMinutes :: i8
contactLength :: double
potentiostat1SerialPort :: string
potentiostat2SerialPort :: string
pickDuty2 :: i8
dropDuty2 :: i8
stepsPerMicroLiter :: double
millimeterPerMilliliter :: double
bufferValveIndex :: i8
sampleValveIndex :: i8
cleanValveIndex :: i8
airValveIndex :: i8
pumpSerialPort :: string
pipetteOffsetX :: double
pipetteOffsetY :: double
mode :: OperationMode
enum OperationMode
SINGLE_LEFT = 0
DUAL = 1
SINGLE_RIGHT = 2
struct SamplerState
timestamp :: i64
laserState :: bool
signal :: Signalization
sampler :: SamplerSettings
protocols :: list<ProtocolLink>
reports :: list<ReportLink>
workSpacesChanged :: bool
methodsChanged :: bool
potentiostatsChanged :: bool
usedHolderCells :: list<i32>
reportUUID :: string
actualScreenShot :: string
protocolError :: bool
endStopStateEngine10 :: bool
endStopStateEngine11 :: bool
endStopStateEngine20 :: bool
endStopStateEngine21 :: bool
endStopStateEngine30 :: bool
endStopStateEngine31 :: bool
endStopStateEngine40 :: bool
endStopStateEngine41 :: bool
actualUnitEngine1 :: double
actualUnitEngine2 :: double
actualUnitEngine3 :: double
actualUnitEngine4 :: double
theoreticalUnitEngine1 :: double
theoreticalUnitEngine2 :: double
theoreticalUnitEngine3 :: double
theoreticalUnitEngine4 :: double
recievedSerialData :: map<string, list<binary>>
enginesEnabled :: bool
ready :: bool
enum Signalization
NONE = 0
SUCESS = 1
ERROR = 2
struct ProtocolLink
name :: string
uuid :: string
boundProtocol :: string
date :: i64
struct ReportLink
name :: string
uuid :: string
date :: i64
protocolUuid :: string
struct UART
busName :: string
path :: string
baudRate :: i32
parity :: Parity
dataBits :: i32
stopBits :: i32
blocking :: bool
enum Parity
None = 0
Odd = 1
Even = 2
Mark = 3
Space = 4
struct Report
name :: string
uuid :: string
date :: i64
protocolUuid :: string
cellToMeasurement :: map<i32, string>
cellToValidity :: map<i32, bool>
cellToPhotoInit :: map<i32, string>
cellToPhotoPost :: map<i32, string>
struct Potentiostat
name :: string
description :: string
deviceSerial :: string
ready :: bool
error :: bool
struct Pump
uuid :: string
name :: string
description :: string
address :: i32
valves :: list<Valve>
units :: i32
unitsPerMicrolitre :: double
pumpSerial :: string
actualUnit :: i32
ready :: bool
inError :: bool
sensorUpperLimit :: double
sensorLowerLimit :: double
struct Valve
id :: i16
description :: string
selected :: bool

Fig. S3. The station software blueprint. From the left, dashed rectangle contains function definitions of a service application programming interface (API). This encompasses all the functionality the device is capable of. That is storage and retrieval of protocols and its results, definitions of potentiostat electrochemical methods, global settings and states, workspace layout definitions and other testing routines such as commands for stepper engines, serial communication with other components and camera image acquisition. Following the arrows, the structures and nesting of data model are depicted. From top to bottom, definitions of structures used for potentiostat measurements and techniques are shown. Next come the definitions of protocol and workspace structures, these govern the physical layout of plates and electrode holders, methods to be carried out on selected cells and other behavior such as electrode pick and drop modes, amounts of sample and buffer to be filled into cells and other parameters. Following structures are part of global settings and machine state, these mostly consist of physical parameters regarding engine control, tool offsets, clamp servos control duties peripheral serial port assignment and operation mode. Machine state contains links to stored protocols and reports, describes actual stepper engine positions, ongoing serial data transactions and other internal flags. Next structures describe serial port parameters and framing definitions, EmStat potentiostat settings and measurement report that contains binding from cells to potentiostat measurement results, photos and evaluation validity. Last structure is used for setting up Kloehn VersaPump syringe dispenser module that is used for dosing chemicals into plate cells.

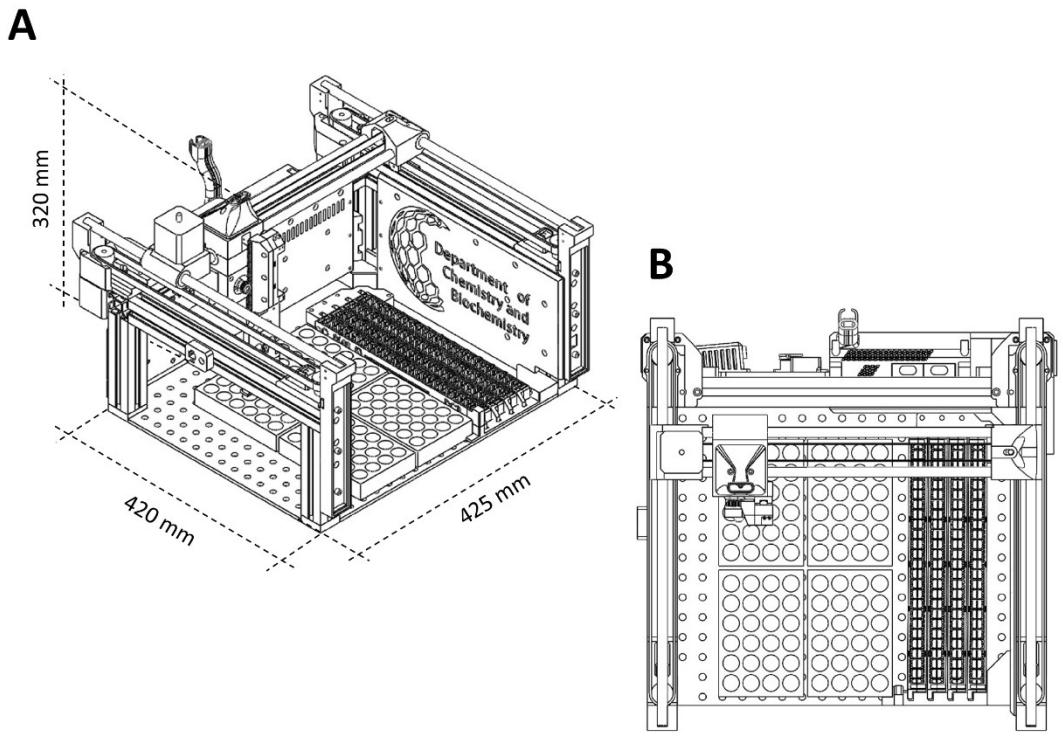


Fig. S4. Blueprint of the reported electrochemical station. (A) Perspective. (B) Top view.

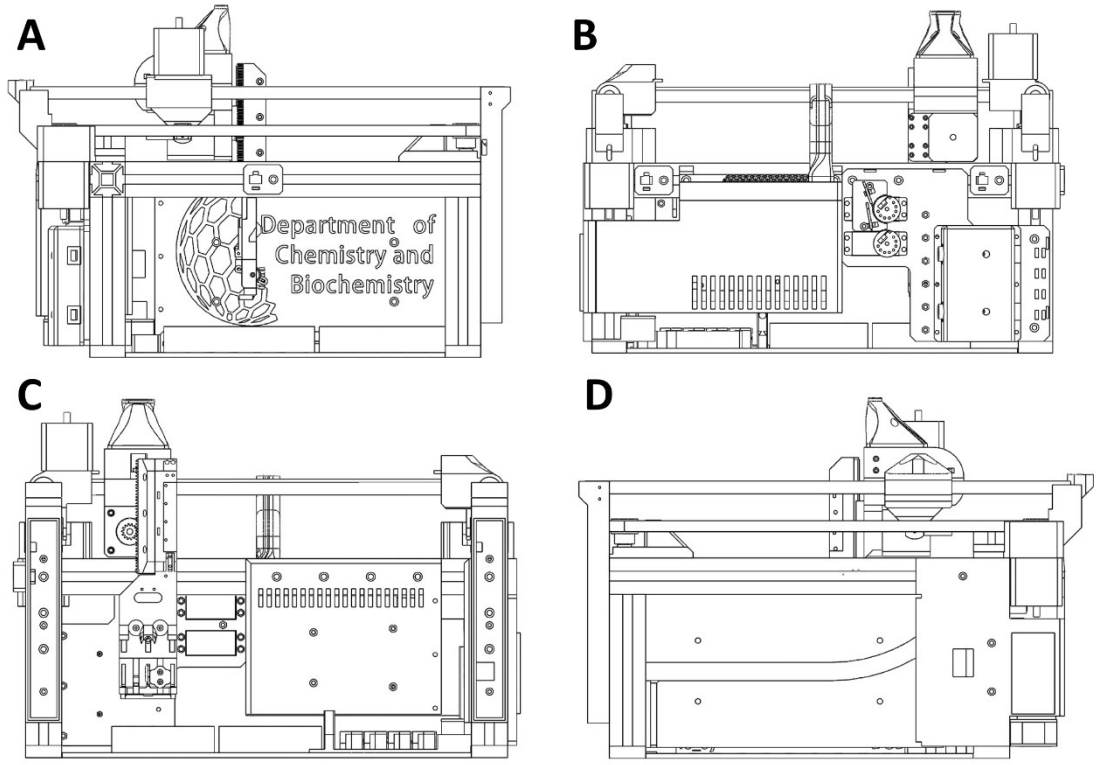


Fig. S5. Blueprint of the reported electrochemical station. (A) Left side view. (B) Right side view. (C) Front elevation. (D) Back elevation.