UNIVERSITY OF CINCINNATI

Date: 2 October 2008

I, Eric MacKnight

hereby submit this work as part of the requirements for the degree of:
Master of Science

in:
Electrical Engineering

It is entitled:
An Autonomous, On-Site Sampling / Analyzing System for Measuring
Heavy Metal Ions in Ground Water

This work and its defense approved by:

Chair: Dr. Chong H Ahn
       Dr. Paul Bishop
       Dr. Joseph Nevin
       Dr. Am Jang
An Autonomous, On-Site Sampling / Analyzing System for Measuring Heavy Metal Ions in Ground Water

A thesis submitted to the

Division of Research and Advanced Studies of

the University of Cincinnati

In partial fulfillment of the requirements for the degree of

Master of Science

In the Department of Electrical and Computer Engineering of

the College of Engineering

2008

by

Eric MacKnight

B.S., Ohio University, 2004

Committee Chair : Dr. Chong H. Ahn
ABSTRACT

The object of this thesis is to develop an autonomous on-site sampling system with electrochemical detection of heavy metals in ground water. The measuring system is comprised of three layers of components: an electrochemical control layer, a fluidic control layer, and a control interface layer. The electrochemical control layer consists of a commercial off-the-shelf (COTS) electrochemical potentiostat, disposable sensor chips, and the circuit board associated with them. The fluidic control layer consists of a pump, valves, polycarbonate microfluidic motherboard with the microfluidic channels etched in, and a circuit board for associated components. The control interface layer consists of a software program written in LabVIEW development environment, a data acquisition card (DAQ), and wireless components. The control interface layer works with both other layers to make the control strategy complete.

The autonomous system was developed in coordination with the development of new disposable Bismuth electrochemical sensors. Bismuth electrodes have the advantage of being more environmentally friendly than traditional Mercury drop electrodes, while maintaining similar sensitivity and other desirable characteristics. The system is approximately 8” W x 11” L (roughly the size of notebook paper) and about 3” deep. The small size and wireless computer interface gives the advantage of being portable for field use while not sacrificing portability for accuracy of measurement. The developed sampling system was fully characterized for sampling and measuring functions in sequence for the analysis of heavy metals from both ground and surface water.
ACKNOWLEDGEMENT

First, I would like to thank the Lord for everything that He has provided me and all the opportunities He has given me in my life. Second, I would like to thank from the bottom of my heart my advisor, Dr. Chong H. Ahn for his encouragement, faith, inspiration, and guidance, and support throughout my graduate study. Without Dr. Ahn's patience, support, and helpful guidance, none of this would have been possible. He has been my role model in research and has so many wonderful things to offer the world in his generous philanthropies and volunteer work. I would also like to thank Zhiwei Zou, Pei Ming Wu, and Jaephil Do for their brilliant contributions and hard work in this group project. Zhiwei's help, most notably with the development of the Bismuth sensor itself, and Pei Ming's support with the circuit board design and fabrication molded the project into what it is today. Jaephil Do was the project manager and assisted in keeping all project time lines on-time and all group members communicating. I would also like to thank all the members of Microsystems and BioMEMS Lab for their tremendous academic and emotional support. Special thanks to Jeff Simkins who has made many custom modifications to various project components, Dr. Kees Van Zelzen (Palmsens BV) for his above-and-beyond support and contributions to the software development, and the National Instruments support engineers for their ongoing support of this project. Finally, I would like to express my deep thanks to my mother, father, and fiancee (who at times acted as project assistant!) for their support, love and prayers.
CHAPTER 1. INTRODUCTION ........................................................................................................5
  1-1. Electrochemical Sensing for Heavy Metal Ion Detection........................................5
  1-2. Research Objective ........................................................................................................7
  1-3. Previous Work Review ..................................................................................................10
  1-4. New Approach ...........................................................................................................11
  1-5. Outline of Thesis .........................................................................................................12

CHAPTER 2. ELECTROCHEMICAL CONTROL LAYER ............................................................13
  2-1. Bismuth Sensor Chip ..................................................................................................13
  2-2. Palmsens Potentiostat ...............................................................................................14
  2-3. Electrochemical Control Circuit Board .......................................................................15

CHAPTER 3. FLUIDIC CONTROL LAYER .............................................................................19
  3-1. Microfluidic Motherboard and Tubing .......................................................................21
  3-2. Pump and Valves ......................................................................................................23
  3-3. Fluidic Control Circuit Board ....................................................................................24
CHAPTER 4. CONTROL INTERFACE LAYER

4-1 . LabVIEW Program Development

4-2 . System Integration Components

CHAPTER 5. EXPERIMENTAL RESULTS AND CONCLUSION

5-1 . Experimental Results

5-2 . Conclusion

5-3 . Suggestions for Future Work

REFERENCES

APPENDICES

Appendix A

Appendix B

Appendix C
List of Figures

Fig. 1-1.  Functional representation of the sampling system, from a component point of view
Fig. 1-2.  Electrical representation of the system
Fig. 1-3.  The overview of the fluidic system, with the chips being the electrochemical sensor area
Fig. 1-4.  Photograph of the assembled system in a shoebox-sized enclosure
Fig. 2-1.  Picture of the disposable Bismuth electrochemical sensor
Fig. 2-2.  Functional representation and early pictures of the Bismuth electrochemical sensor
Fig. 2-3.  Picture of the COTS Electrochemical potentiostat
Fig. 2-4.  Circuit schematic of Electrochemical Control Circuit Board
Fig. 2-5.  Close-ups of selected areas of the 3-mux circuit schematic.  “Holes” are through holes for the gold spring electrodes.  The picture on the right shows the contact pads for the potentiostat
Fig. 3-1  Sample 1 is prepared with buffer
Fig. 3-2.  Sample is loaded to sensor 1 and the measurement is made
Fig. 3-3.  Sample 2 is prepared with buffer
Fig. 3-4.  Sample is loaded to sensor 2 and the measurement is made
Fig. 3-5.  Bottom layer of microfluidic motherboard
Fig. 3-6.  Top layer of fluidic motherboard
Fig. 3-7.  APT Peristaltic Pump
Fig. 3-8.  Latching Valves
Fig. 3-9.  Schematic of the fluidic control circuit board schematic
Fig. 3-10.  Enlarged section of the multiplexer and 2 valve circuits on the left and an enlarged section of a valve circuit and the pump circuit on the right
Fig. 4-1.  Main Screen of the Program, Showing General Statuses
Fig. 4-2.  Data Load Screen, Showing the Current Configuration Loaded to the Potentiostat
Fig. 4-3. Set Configuration Screen, Where a New Set of Measurement Parameters Are Loaded

Fig. 4-4. Pump and Valve Settings Screen, Where the Fluidic Control Parameters Are Set

Fig. 4-5. Enclosure from Bud Industries

Fig. 4-6. Photograph of the assembled system in a shoebox-sized enclosure

Fig. 5-1. Result of the three measurements on the same chip (different sensors) and correlation of results. The above graphs are from a 2ppm Cadmium solution.

Fig. 5-2. Again, another 2ppm Cd solution, this time with 3 different sensors to prove reproducibility.

Fig. 5-3. A single curve showing the I-V characteristics from a single sensor. A 0.5ppm Cd sample was used in this case.

Fig. 5-4. All three results from the 0.5ppm Cd solution shown in Fig. 5-2.

Fig. 5-5. A small amount of analyte (0.1ppm) measured also showed good data correlation.

Fig. 5-6. Screenshot from a cyclic voltammetry measurement shows good data for 3 different sensors.

Fig. 5-7. Data Correlation (analyte concentration vs. peak current)

Fig. 5-8. Table of the data plotted in Fig. 5-8.
CHAPTER 1. INTRODUCTION

Electrochemical sensing is a well-developed field, with the first electrochemical sensing being performed in 1922 by Jaroslav Heyrovsky in what is today the Czech Republic. Since, many improvements have been made, especially in the 1960s and 1970s when advances in theory, methodology, and instrumentation allowed for better measurements over wider ranges [1]. Today, several different commercial products exist for purchase.

Bismuth electrodes have the advantage of being non-toxic, as are the Mercury-based electrodes that have been the de facto standard for decades. The method of ion attachment is different, however. With Mercury electrodes, the ions attach themselves all over the surface of the electrode and then are stripped off (assuming stripping voltammetry). With the Bismuth electrodes, the Bismuth forms an alloy of itself and the analyte ion. Microelectrodes have the advantage of having high sensitivity due to the small separation between electrodes that is possible and also from the precise control over electrode dimensions and thicknesses. They are also typically easily mass produced.

1-1. Electrochemical Sensing for Heavy Metal Ion Detection

In voltammetry, the effect of the applied voltage and the resulting behavior of the redox current are described by several known laws. The Nernst equation

\[ E = E^0 - \frac{RT}{nF} \ln \frac{c_R^0}{c_O^0} \]

is the applied potential, the concentrations of the redox species at the electrode surface \( (C_o^0 \)
and $C_R^0$, the rate of the reaction $k^0$, for concentrations of $O$ and $R$ at the surface of the electrode. $R$ is the molar gas constant (8.3144 J mol$^{-1}$ K$^{-1}$), $T$ is absolute temperature, $n$ is the number of electrons transferred, $F$ is the Faraday constant (96,485 C/equiv) and $E^0$ is the standard reduction potential for the redox couple [1]. The Butler-Volmer equation

$$\frac{i}{nFA} = k^0 \left\{ c_O^0 \exp[-\alpha \theta] - c_R^0 \exp[(1 - \alpha) \theta] \right\}$$

where $\theta = nF (E - E^0)/RT$, $k^0$ is the heterogeneous rate constant, $\alpha$ is the transfer coefficient, and $A$ is the area of the electrode. This equation allows for a relationship between $I$ and $k^0$.

Fick's law

$$\Phi = -AD_0 \left( \frac{\partial c_O}{\partial x} \right)$$

applies to cases where the current flow depends directly on the flux of ions to the electrode surface [1]. These three equations are routinely used to analyze the results from a voltammetric measurement. Of the many voltammetric techniques available, square wave voltammetry has the advantages of faster scan rate and increased sensitivity. Anodic stripping voltammetry is widely used for trace metal determination with the ability to determine four to six trace metals simultaneously. [1]

Measuring heavy metals is no different than measuring any other ion in solution. In SWASV, a potential signal represented as a staircase wave added to a square wave is applied over the working electrode, ions attach or implant themselves to the working electrode. As potential is varied, the ions strip off of the electrode and the current is measured passing through the counter electrode. The
current is then plotted against the voltage for later analysis to find the amount of analyte present in the solution.

1-2. Research Objective

The object of this thesis is to develop a new autonomous system to utilize the new, disposable Bismuth electrodes. Design constraints of the system were set for each control layer. For the electrochemical control layer including the electrochemical sensors themselves, the most important design constraints were noise and sensitivity. For the chosen potentiostat, size was also a crucial factor in the choice of the commercially available sensor system. The circuit board was to have multiplexer function for the potentiostat and minimal noise of the circuit is desirable for the minimized distortion of the voltammetric signal. Basically, in order to showcase the performance of the Bismuth sensor, it was devised that the potentiostat and everything between it and the computer was able to characterize its noise and its performance without being limited by the measuring equipment's performance. For the fluidic control layer, sensitivity, reproducibility, and repeatability were the most important constraints of the sensor chips. The purpose of their design was to be disposable, so they also needed to stay true to their environmentally-friendly makeup. Size and prevention of bubble formation were the most important constraints for the microfluidic motherboard and tubing and was also an extremely important characteristic of the pump and valves. The valves and the fluidic circuit board had the additional design constraint of being low power and low noise. The control system interface layer was to be wireless and have low power consumption. As to be seen later, some of these constraints were conflicting, as is typically the case in engineering. The functional diagram of the system is shown below in Fig. 1-1 and the electrical connection of the assembled system is shown in Fig. 1-2. The fluidic system overview is shown in Fig. 1-3 and the photograph of the completed
system is shown in Fig.1-4.

Fig. 1-1. Functional diagram of the sampling system.

Fig. 1-2. Electrical connections of the system.
Fig. 1-3. The overview of the fluidic system, with the chips being the electrochemical sensor area.

Fig. 1-4. Photograph of the assembled system in a shoebox-sized enclosure.
1-3. Previous Work Review

There are many published papers having done similar work to our project with some differences. Kwakye et. al. reported a microcontroller-based system with RS232 communication to a PC of the results[2]. The system is powered by a single AA or AAA battery for 3 hours of life. Our system, in comparison, has much larger battery packs, but also autonomous measurement and sample loading, and longer battery life as well. Similarly, Sheu, et. al. Developed a miniature system based on a microcontroller performing measurements and storing data to later be off-loaded to a PC, in this case with a flash memory card[3].

Yun, et. al. described a miniature low-power wireless remote environment monitoring system for heavy metal ion detection to measure water pollution, which is indeed our same application[4]. They, however, used mercury-drop electrodes to achieve measurement. Their goal also was to have a microcontroller-based system with RF communications back to a base station for data transmission.

Paeschke, et. al. Described a system with sensitivity down to 50 pA [5]. This system uses a microcontroller as well to control the sensing routines. Again, this system used a RS232 communication protocol to ofload data to a PC for further analysis with other data analysis tools. Avdikos, et. al. also chose this method, as did Martin, et. al., Christidi, et. al., Tercier, et. al, and Martin, et. Al [6,7,8,9,10]. Some of these research groups focused on sensitivity, others on portability, and yet others on minimizing physical size. Our system, in contrast, focuses on delivering a package complete with pump, valves, sensors, and controlling software and analysis tools.
1-4. New Approach

The focal point for this project was the Bismuth electrochemical sensors. The rest of the system was to be designed around them. The Bismuth electrochemical sensors were designed to be disposable and with similar performance to Mercury-based electrodes. The purpose of the Bismuth electrode was to achieve a disposable chip with less toxicity and environmental impact than conventional electrochemical sensors. So, the Bismuth electrochemical sensors were designed and characterized as a disposable platform with similar performance to Mercury-based electrodes.

Bismuth (Bi), which has much less toxicity while maintaining the similar electrochemical performance compared with Hg, is an alternative working electrode material for the ASV of heavy metal. Bi is an environmentally-friendly element, with very low toxicity, and a widespread pharmaceutical use. “While amalgam formation is responsible for the stripping performance of Hg electrodes, the attractive and unique behavior of Bi electrode is attributed to the formation of multi-component Bi alloys with numerous heavy metals.” [11] Adding Bi ions into the sample, and plating of Bi ions before or with target metals together on a carbon electrode is the most commonly used method to prepare the Bismuth electrode and has shown good sensitivity for trace metal detection. This method can be undesirable for the in on-site and online trace metal measurement in natural- or bio-environment because it introduces some additional Bi ions: The additional Bismuth can react with certain analytes and it may also be harmful or contaminate the media (e.g. cell culture media). The Bismuth electrodes also cannot be used for the continuous long term monitoring and the pH of the sample has to be controlled below 5 to avoid the hydrolysis of Bi (III) ions which is not practical for many biological and clinical applications.
The development of an autonomous and on-site monitoring system is very desirable for monitoring heavy metal ions in ground water, using the Bismuth electrode-based electrochemical sensors. The system that is presented is fully autonomous and portable, which has the advantages of saving money in the form of time and manpower, reducing the risk of contamination since it is done in the field, and delivers quicker results than conventional measurements. For simplicity in design, the control system was logically divided into three parts, or layers. The first of these is the electrochemical control layer consisting of a commercial off-the-shelf (COTS) electrochemical potentiostat, disposable electrochemical sensor chips, and the circuit board associated with them. The fluidic control layer, consists of the pump, valves, polycarbonate microfluidic motherboard, and the circuit board for associated components. The control interface layer consists of a software program written in LabVIEW development environment, a data acquisition card (DAQ), wireless components, and the packaging. The control interface layer works with both other layers to allow the controlled communication between them.

1-5. Outline of Thesis

The research objective of this thesis, some previous work and our new approach are described in Chapter 1. In Chapter 2, the electrochemical control layer is presented. In Chapter 3, the fluidic control layer is presented. Chapter 4 introduces the control interface layer, combining the components from Chapters 2 and 3. Chapter 5 consists of the conclusion and contribution of this research, and also suggestions for future development.
CHAPTER 2. ELECTROCHEMICAL CONTROL LAYER

The electrochemical control layer consists of the Bismuth electrochemical sensor itself, upon which the system was designed, the potentiostat, and the supporting electrical circuitry. This part of the system is responsible for measuring the amount of heavy metal analyte is present in the ground water.

2-1. Bismuth Electrode

A disposable Bismuth sensor chip was developed for the reasons stated in Chapter 1. The Bismuth electrochemical sensors were designed to be disposable and with similar performance to Mercury-based electrodes. It had dimensions 4.5 x 2.5 x 0.2 cm and features gold contact pads, a Ag/AgCl Reference electrode, a Bismuth working electrode, and a gold counter electrode as shown in Figure 2-1. The functional diagram of the Bismuth electrochemical sensor and measurement method is shown in Fig. 2-2. The fabrication and details of the sensor chip itself is another research topic of its own. More information about the fabrication and sensor characteristics is found in reference [11].
2-2. Potentiostat

To save time and money in development of the system, it was decided to purchase an electrochemical potentiostat that was small, portable, sensitive, and with the capability for many different types of electrochemical measurement. The perfect fit for our system was the commercially-developed system Palmsens (Palmsens Instruments, Netherlands BV, approx. EU3,000) as shown in Figure 2-3. Its data sheet states that it has a voltage range of 2 V to -2 V (1 mV resolution) with current measuring range from 1 nA to 10 mA (0.1% resolution of the current range) and accuracy of 0.2% of the current range from 100 nA to 10 mA. The high resolution of the Palmsens potentiostat instrument gives a 1 pA resolution in the lowest (1 nA) measuring range. Palmsens also boasts serial or Bluetooth communication options and 9 different measurement techniques (all custom programmable). The available measurement techniques were differential pulse, square wave (stripping) voltammetry, normal pulse, ac voltammetry, stripping chronopotentiometry or PSA, amperometric and pulsed amperometric detection, linear sweep and cyclic voltammetry. This work
only implemented the square wave (stripping) voltammetry and the cyclic voltammetry.

Fig. 2-3.  Picture of the COTS electrochemical potentiostat.

Palmsens has dimensions of 155 mm x 85 mm x 35 mm, small enough for our application, has internal batteries for 6-8 hours of operation, and weighs only 0.45 kg. With all these features and performance, it was decided to use this device as the potentiostat for our measuring system. Although Palmsens is designed to be a ready-to-use system, since we required a sophisticated autonomous system, we decided not to use the Palmscan or Palmtime software included in the package and to instead develop our own. The software development for the Palmsens driver is discussed in section 4-1. LabVIEW Development Program. The Palmsens also has as an available option, a multiplexer for use with it. However, we decided to develop our own multiplexer due to the limitations it imposed on us. The multiplexer system we designed is the subject of section 2-3. Electrochemical Control Circuit Board.

2-3. Electrochemical Control Circuit Board

As mentioned in section 2.2, the Palmsens multiplexer had some limitations, so a new multiplexing system was designed and fabricated. One of the system requirements was that there was to be 21 electrochemical sensors on the system, each with separate counter and reference electrodes.
Palmsens, at the time of system design, only offered a multiplexer for 8 sensors, with common counter and reference electrodes. Therefore, it was determined to design a custom multiplexing circuit board. Some type of circuitry would have still needed to exist either way for electrical contact with the sensors and the microfluidic system.

The new design requirements of the circuit board were for it to add no significant noise to what the Palmsens system had already. Palmsens was taken to open circuit potential for all 4 connectors (working, reference, and counter electrodes and ground as well). It was shown repeatedly that the maximum noise of the system was approximately 1 nA at open circuit. Therefore, the goal was set that with the multiplexing system in place, an open circuit noise of approximately 1nA should be achieved.

In order to meet the design requirements, some of the best commercially available low noise IC chips and device components should be chosen. Since this system was to be hand-soldered onto a circuit board, it was best to use DIP packaged chips, as opposed to TQFP or other smaller electronics that may have slightly better performance. Since a simple multiplexer functionality for 3 (three) 21:1 multiplexers, allowing all current and voltage ranges from the circuitry to be echoed on the multiplexer lines was needed, three 2:1 multiplexers and six 16:1 multiplexers (one for each electrode connection – WE, CE, RE) were purchased. A 2:1 multiplexer in series with two 16:1 multiplexers was chosen instead of a 32:1 multiplexer because of the large size of all 32:1 DIP multiplexers and the difficulty of avoiding overlapping the metal traces on the printed circuit board. There is more noise when having more multiplexers in series and also in having more possibilities for failure, but to keep a compact size, and also due to the limited number of DIP 32:1 multiplexers being available, it was a trade-off we decided was acceptable.
The MAX 309CPE (Maxim IC, USA) and DG406EWI (Maxim IC, USA) were used in series to achieve 32:1 multiplexer functionality. The MAX 309 was chosen for its low noise and ability to pass bipolar signals across itself. Similarly, the DG406 was chosen for its low noise and range, but the SOIC version was used because of the DIP chip's large size and the limited “real estate” on the PCB board. The MAX309 is a 4:1 MUX that we short-circuited its two inputs to force 2:1 functionality, so whereas a normal 2:1 multiplexer would receive a binary 1 or 0 signal, our multiplexer has 2 input address bits that give a 00 or 11 in binary. We did this instead of choosing a 2:1 multiplexer because of the lack of quality 2:1 analog multiplexers that met our requirements. A picture of the circuit is shown below in Figs. 2-4 and 5.

The DG406EWI has an input leakage of 10 nA maximum (1nA under normal conditions), a signal range of +/- 15V, a low on-resistance of approximately 60 ohms, and can pass 30 mA of current through it. The MAX309 has an input leakage of 40 nA maximum (<1nA under normal conditions), a signal range of +/- 15V, a low on-resistance of approximately 60 ohms, and can pass 30 mA of current through it.
Fig. 2-4. Circuit schematic of Electrochemical Control Circuit Board

Fig. 2-5. Close-ups of selected areas of the 3-mux circuit schematic. “Holes” are through holes for the gold spring electrodes. The picture on the right shows the contact pads for the potentiostat
CHAPTER 3. FLUIDIC CONTROL LAYER

The fluidic control layer consists of the pump, valves to direct the fluid flow, polycarbonate microfluidic motherboard, and the supporting electrical circuitry. This part of the system is responsible for directing the sample water to the appropriate measurement area while minimizing bubble formation and bubble presence. The scheme of the fluidic flow is demonstrated below in Figs. 3-1 to 3-4. These figures show the system design from a fluidic flow point of view.

Fig. 3-1 Sample 1 is prepared with buffer.
Fig. 3-2. Sample is loaded to sensor 1 and the measurement is made.

Fig. 3-3. Sample 2 is prepared with buffer.
3-1. Microfluidic Motherboard and Tubing

The microfluidic motherboard was designed with the requirements of minimizing bubble formation and contamination, having a hydrophobic surface, and properly directing the fluid flow. Polycarbonate was chosen as the substrate material for its hydrophobic surface and ease of modification by normal machining methods. The polycarbonate slab was purchased from MSC Industrial Supply Co., USA and designs were made in autoCAD (Autodesk, USA); the machining was then carried out by Kenco Mold, Cincinnati, OH. The design of the polycarbonate microfluidic motherboard is described in Figs. 3-5 and 6.
Fig. 3-5. Bottom layer of microfluidic motherboard.

Fig. 3-6. Top layer of fluidic motherboard.
The hard PEEK tubing was purchased from Upchurch Scientific and the soft tubing from Fisher Scientific. It was found that the first type of soft tubing we had used was causing bubble formation and contaminants were sticking to the interior surface of the tubing. The new Silicone tubing fixed this problem and the epoxy seal around the connection prevented air bubbles and secured the tubing to the microfluidic motherboard.

The disposable electrochemical sensing chips were held air-tight to the microfluidic motherboard by a system of rare-earth neodymium-iron magnets. These magnets (Amazing Magnets, USA) were physically inserted into the plate in Fig. 3-6 and also into a top cover plate to ensure a strong fastening.

3-2. Pump and Valves

A pump was necessary to transport the sample fluid from the sample tubing to the electrochemical measurement area. The SP100VC peristaltic micropump (APT Instruments, USA) was chosen for several reasons. The compact size, DC-driven motor, variable flow rate, hydrophobic surface of contacting materials, and low power consumption led to the choice of the SP100VC. The pump takes 12 VDC and 100 mA under 100 % duty cycle. As described later, the duty cycle was adjusted in the LabVIEW software program. The maximum flow rate of the entire system with the pump, tubes, microfluidic motherboard, and valves all connected was 3.25 mL/min. The pump measured 84 x 140 x 97 mm in size and weighs 650 grams as shown in Fig. 3-7.
The valves chosen to control the direction of the fluid flow were the model LHLX0500200B (The Lee Co, USA). These valves were chosen primarily for their small size and very low power consumption. The valves are cylindrical in shape and have a length of 32 mm and a diameter of 7.5 mm as shown in Fig. 3-8. The low power consumption is due to the latching action of the internal relay that controls the valve's fluidic state. The valve consumes 40 mA at 12 VDC for a minimum 10 msec pulse for each time it switches. There is no visible leakage from the input port to the unselected port when the valve is completely sealed with its face to the microfluidic motherboard.

3-3. Fluidic Control Circuit Board

To function properly, the valves and pump have to be connected to some circuitry to properly take the control signal from the computer hardware (controlled by the LabVIEW program) and interface it to the pump and valves. The overall circuit theme was to use transistors as signal
amplifiers and use analog multiplexers to direct the control signal to the proper valve and get that selector signal from the DAQ card described in Chapter 4.

The pump circuit is very simple, consisting of a basic amplification circuit using a TIP121 transistor with a diode to protect the transistor and a resistor to protect the transistor base. The 5 V control input signal is amplified into a 12 VDC signal that the pump requires for operation.

The valve circuit had the added complexity that in order to latch the valve back and forth, the electrical leg of the valve that gets the input pulse signal must be switched. An analog switch was chosen for this purpose. The ADG436 analog switch (Analog Devices, USA) was used for its signal range and low leakage current (<5 nA maximum). The +/-15 VDC input voltage range covers the +/-12 VDC signal used.

The Fluidic Control Circuit Board is shown in Fig. 3-9 and 10. The selected valve was controlled by the same analog multiplexer used on the sensor control circuit board DG406CPE, only this time a DIP version was used for ease of use since only one was needed here.

Fig. 3-9. Schematic of the fluidic control circuit board.
Fig. 3-10. Enlarged section of the multiplexer and 2 valve circuits on the left and an enlarged section of a valve circuit and the pump circuit on the right.
CHAPTER 4. CONTROL INTERFACE LAYER

The control interface layer consists of the LabVIEW (National Instruments, USA) program along with the supporting components, like the DAQ card, power converters, power supplies, communication equipment, container, and equipment to secure all components. The purpose of this layer is to tie all system components into one streamlined system. From a system point of view, this is the most important layer. Indeed, most of the labor in developing this research project went into developing the LabVIEW software program and ensuring reliable control over system components, as would be expected by the user.

4-1. LabVIEW Program Development

The control logic and user interface for the control system was developed using a LabVIEW program written from scratch in LabVIEW 8.2 development environment. The program was written with the user in mind and therefore was designed to be as simple as possible, without sacrificing the necessary amount of control. The LabVIEW device driver was written for the Palmsens instrument and was thoroughly tested for system stability. The communication protocol used for communicating with the Palmsens is RS232, which was later converted to Bluetooth 2.0. The LabVIEW program was written in pieces, or subroutines, and then integrated into a main program.

The main screen interrogates Palmsens every 3 seconds and reports such data as temperature, voltage, current, and all error and status messages delivered by Palmsens. The user is also allowed to select the COM port for communication with Palmsens for portability to other machines and ease of integration to several Bluetooth device manufacturers.
Fig. 4-1. Main Screen of the Program, Showing General Statuses.

Fig. 4-2. Data Load Screen, Showing the Current Configuration Loaded to the Potentiostat.
Fig. 4-3. Set Configuration Screen, Where a New Set of Measurement Parameters Are Loaded.

The Palmsens driver was developed using information about the communication standard from the Palmsens inventor, Dr. Kees Van Zelzen. Screenshots of the software programs are shown in Figs. 4-1 through 4-3. The driver includes the ability to send and receive measurement method commands to Palmsens, to command it to make a measurement with the current method, to receive and decrypt that data, and to receive and decode all status messages from Palmsens. The raw data from each measurement is saved to the C:\ directory with appropriate filenames and data stored in text format, easily exportable to Microsoft Excel, Lotus 1-2-3, or similar for analysis. By default and for ease of use, the software package also includes its own analysis of the data. The raw data is fed through a Sovitzky-Golay lowpass filter and plotted, with the user left to draw tangent lines so that the current peak magnitude from this baseline may be measured and voltage at which it occurs may be recorded.
The control logic for the pump and valve control was then written by using a USB-6008 (National Instruments, USA) data acquisition card (DAQ card). This card was chosen for its affordability in meeting our design requirements. A very flexible user interface was used and the software was developed as such to allow the user to set pumping times, measuring times, and the order to measure which samples, if desired. The system is designed by default to measure each sample three times and report the results to files saved on the C:\ drive, but is not limited to such actions. The pump and valve control screen also allows the user to enter separate pumping times and speeds for the sample and for the buffer washing step as shown in Fig. 4-4. The user may also select an “AUTO” function to have the program automatically step through the entire control scheme in the case that either the control strategy is set for a long time or the user does not wish to see the results right away, but instead to view the resulting files at a later date at his or her convenience.

Fig. 4-4. Pump and Valve Settings Screen, Where the Fluidic Control Parameters Are Set.
The LabVIEW program was made with the user in mind and robustness was the ultimate goal. Care was taken to account for all missteps by the user to not affect the system negatively. Many optional data analysis screens were added for the user's convenience.

4-2. System Integration Components

The data acquisition card was utilized with its digital outputs geared to be used for address section on the multiplexers described in the first two sections and for the analog switch as well. The first analog output was used to generate the pump signal duty cycle and the second one is used for the valve pulse signal used to latch the valves back and forth, open and closed, depending on the switch selector digital output bit. No inputs were used on the data acquisition card.

The ASD30-12D5 (Astrodyne, USA) power converter was used in conjunction with MH-P1012V3500-C01 battery pack (Batteryspace.com). The ASD30 supplies the +/- 12VDC that the multiplexers, valves, pump, switches, and other circuit components need to operate. The 5VDC is supplied to provide power to the Wireless USB system to be described later. The ASD30 is a very robust power converter, able to withstand undesirable conditions the DAQ card presents without malfunctioning. The battery pack is able to provide 3500 mAh, allowing the system to function for over 10 hours of operation with the wireless system enabled and over 1 day without the wireless system.

The system has the option of being wired or wireless. The original system was built with USB cable and RS232 cable going from the system package in the box to the PC. The system was then upgraded to incorporate Wireless USB and Bluetooth. The Bluetooth was used to communicate
between the Palmsens and the PC. It was included in the sale of the Palmsens and operates without drawing power from the main system power – the PC and Palmsens both have their own battery power supplies. The wireless USB was used to connect the USB-6008 DAQ card to the PC without restrictions on wire length. IOGear's (IO Gear, USA) GUWH104KIT contains the transmitter and receiver for our wireless USB communications.

The last part of our Control Interface Layer is the container all the components reside in. We chose a notebook-size container from Bud Industries as shown in Fig. 4-5. The container was originally designed to meet NEMA 4X requirements, but due to space limitations and changing constraints after components were purchased, the box was necessarily modified to accommodate such changes and therefore now no longer meets the NEMA standard. The sampling and monitoring system developed in this work is shown in Fig. 4-6.

![Fig. 4-5. Enclosure from Bud Industries](image)
CHAPTER 5. EXPERIMENTAL RESULTS AND CONCLUSION

5-1. Experimental Results

The results of the system characterization were plotted and displayed in the LabVIEW user interface. Many different solutions of different heavy metals were measured and displayed with different sensors and with different electrode sets on the same sensor. Fig. 5-1 show the resulting data from a square-wave stripping voltammetry (SWSV) measurement for 3 different sensors on the same chip using the same solution. The excellent correlation between maximum current, and the voltage at maximum current is shown. The maximum current indicates the amount of analyte present and the voltage where the peak current is located at indicates which analyte it is. Fig. 5-2 shows the resulting
data from the same solution of Cadmium with a different chip. It indicates the excellent reproducibility of results within different chips when compared to Fig. 5-1, and also excellent reproducibility between its own sensors.

Fig. 5-1. Result of the three measurements on the same chip (different sensors) and correlation of results. The above graphs are from a 2ppm Cadmium solution.
Fig. 5-2. Again, another 2ppm Cd solution, this time with 3 different sensors to prove reproducibility

In Fig. 5-3, the I-V characteristics of one of the sensors is shown. The white line shows the sensor data received and the red line is a “baseline” off of which the peak current is measured. The smoothness of the line is not due to a filtering function completely, but is mostly due to the data points following a very nicely curved line. A 500 ppb solution of Cadmium was used here to also show the sensitivity of the system to pick up such small concentrations of analyte. Fig. 5-4 shows a different set of 3 sensors measuring the same 500 ppb solution as in Fig. 5-3. There is again excellent correlation between the different sensors on the same chip, even with the lower solution concentration.
Fig. 5-3. A single curve showing the I-V characteristics from a single sensor. A 0.5ppm Cd sample was used in this case.

Fig. 5-4. All three results from the 0.5ppm Cd solution shown in Fig. 5-2.
Fig. 5.5 shows the data for 3 sensors on the same chip with a very low concentration (100 ppb). Similarly good results for reproducibility are shown here.

![Data correlation diagram]

**Fig. 5-5.** A small amount of analyte (0.1ppm) measured also showed good data correlation

Fig. 5-6 shows a different method of voltammetric measurement. The demonstrated measurement in the figure is cyclic voltammetry, while the other results in previous figures were of square wave stripping voltammetry. Good correlation is seen among 3 different sensors on the same chip for this alternative measurement method as well. The overall data correlation for a set of SWSV measurements is shown in Fig. 5-7 and the plot shows great linearity. The table in Fig. 5-8 shows the data used to construct Fig. 5-7.
Fig. 5-6. Screenshot from a cyclic voltammetry measurement shows good data for 3 different sensors.

\[ f(x) = 0.02x + 12.39 \]
\[ R^2 = 0.97 \]

Fig.5-7. Data Correlation (analyte concentration vs. peak current)
The fluidic control layer performed very well also, with the fluid being directed to the commanded path repeatedly as expected. There were greater than one hundred correct executions of changing the fluidic path observed.

5-2. Conclusion

In this work, a portable, autonomous system was designed for the measurement of heavy metal ions in ground water sample solution. The system performed well at its demonstration, to the satisfaction of the group for whom the project was designed. The system is able to accurately control the direction of fluid flow through the correct microfluidic path repeatedly and measure the sample with the correct electrodes without bubble interference or overwhelming noise from circuit or other components. This concept was proven and the results were satisfactory to most persons involved with the project. The data showed a linear increase in peak current with increase in concentration. The correlation was excellent, with $R^2 = 0.97$.

5-3. Suggestions for Future Work

A more robust circuit board should be used with the improvements we decided would help
stabilize the system from leaving some parts of the unselected components at unknown voltage states. The new circuit board should have power indicators to ensure power is in the appropriate ranges and that overcurrents do not occur. The overcurrents (that are able to be controlled in the current system) are caused by the NI USB-6008 DAQ card short-circuiting its input/outputs to ground potential in the case that an applied voltage is not expected. If power is applied to the DAQ before it is communicating successfully with the computer, then this short-to-ground condition occurs. The circuit board was made to prove that the concept works and prove the system operation. It was hand-soldered by the author and could have much better contacts. The circuit components, like multiplexers, switches, etc. could be replaced with smaller versions to compact the size of the circuit boards if that is desired. Most chips on the circuit board were DIP and the layout was designed with prototyping in mind.

The computer could be replaced with a microcontroller with appropriate processing power and data downloading interface. The valves used have some problems with longevity and fail too often. They could be replaced with a different type of valve with similar functionality.
REFERENCES


APPENDICES

Appendix A

Palmsens Protocol Manual
Communications Protocol
Contents

1. PalmSens communication protocol .................................................................................................................................................................................. 3
   1.1 General commands ......................................................................................................................................................................................... 3
   1.2 Receiving from PalmSens ...................................................................................................................................................................................... 3
      1.2.1 Receive method from PalmSens .............................................................................................................................................................. 4
      1.2.2 Receiving data from PalmSens ............................................................................................................................................................... 4
      1.2.3 Some examples .......................................................................................................................................................................................... 5

2. Remote control of PalmSens ....................................................................................................................................................................................... 6
   2.1 Fast remote control mode ...................................................................................................................................................................................... 6
   2.2 Set potential or current ..................................................................................................................................................................................... 6
      2.2.1 Set potential of WE2 ........................................................................................................................................................................... 6
      2.2.2 Set analog output on miniDIN connector .............................................................................................................................................. 7
      2.2.3 Set mode of potentiostat/galvanostat and current range .......................................................................................................................... 7
   2.3 Reading the current of WE2 .............................................................................................................................................................................. 8
   2.4 Offset of BiPotentiostat .................................................................................................................................................................................... 8
   2.5 Read current, potential or auxiliary analog input signal ................................................................................................................................... 10
   2.6 Read current, potential or auxiliary analog input signal continuously at fixed rate .................................................................................. 10
   2.7 Set digital output lines of miniDIN connector ......................................................................................................................................... 10
   2.8 Read digital input line of miniDIN connector ............................................................................................................................................ 11
   2.9 Sending and receiving integer values as ASCII characters ....................................................................................................................... 11
      2.9.1 Sending an integer to PalmSens in the range of 0 to 255 is done as follows: .................................................................................. 11
   2.10 Receiving PalmSens versions and power status ..................................................................................................................................... 11

3. Method parameters ............................................................................................................................................................................................... 12

Appendix A: Manual control flow chart ........................................................................................................................................................................... 14

June 2005
1. PalmSens communication protocol.

PalmSens is connected to a PC by means of a standard null-modem cable. The communication port settings are: 57600,N,8,1 with No handshaking. Communication is done by sending ASCII characters. A sample program (PSControl) written in Visual Basic is available on request.

1.1 General commands

PalmSens is controlled by the PC using the following characters or commands:

- ‘L’ (see Ch 3.) Load PalmSens with method parameters from PC.
- ‘M’ Start measurement using the loaded method.
- ‘Y’ Skip current pretreatment stage, i.e. conditioning, deposition or equilibration stage. This command is only relevant when a measurement has been started.
- ‘Z’ Abort current measurement.
- ‘s’ Send measured data.
- ‘K’ Send hexadecimal values of the specified method to PC.
- ‘k’ Display hexadecimal values of method on LCD display. This is only useful during debugging application software.
- ‘N’ Calibrate PalmSens analog electronics.
- ‘J’ (see Ch 2.1) Set PalmSens in the ‘Fast Remote Control’ mode
- ‘j’ (see Ch 2.1) Set PalmSens in the ‘Normal Remote Control’ mode.
- ‘c’ Used for remote control. Available remote control commands are:
  - ‘d’ (see Ch 2.2) Set potential of current
  - ‘G’ (see Ch 2.2.3) Set mode of potentiostat/galvanostat and current range
  - ‘W’ (see Ch 2.2.4) Set current range of the bipotentiostat (optionally available module)
  - ‘d’ (see Ch 2.2.1) Set potential of the bipotentiostat (if available) or analog output signal on miniDIN connector
  - ‘a’ (see Ch 2.5) Read current, potential or analog input signal on miniDIN connector
  - ‘b’ (see Ch 2.6) Read current, potential or analog input signal on miniDIN connector at a fixed rate.
  - ‘v’ (see Ch 2.7) Set digital output lines of the miniDIN connector
  - ‘r’ (see Ch 2.8) Read digital input line of miniDIN connector

1.2 Receiving from PalmSens

PalmSens sends information preceded by the following characters:

- ‘r’ (see Ch 2.10) PalmSens sends it’s hardware version, software (firmware) version and power status.
- ‘L’ (see Ch 3.) PalmSens expects 66 bytes with the method parameters. Each byte has value from 0 to 255.
- ‘K’ (see Ch 3.) PalmSens sends 66 bytes with the method parameters. Each value is within the range 0 to 255. The information is terminated with an End-Of-File sign ‘*’.
- ‘S’ (see Ch 1.2.2) The complete curve of the most recent measurement is sent. This is normally done immediately after a measurement but also after receiving the command from the PC ‘s’.
- ‘U’ (see Ch 1.2.2) A single data point is sent: potential, current and status byte.
- ‘T’ (see Ch 1.2.2) Measured values of potential, current, status byte, temperature and noise are sent. These values are measured during idle stage, or conditioning, deposition and equilibration stages.
- ‘V’ (see Ch 1.2.2) An amperometric curve is sent. Each data point is preceded by ‘Y’.
- ‘Y’ (see Ch 1.2.2) A single chronoamperometric data point follows.
- ‘N’ This character is followed by 12 bytes with the calibration results.
- ‘*’ End-of-file sign after sending a measured curve.
1.2.1 Receive method from PalmSens
The command ‘K’ can be used to download a method from PalmSens and store it on the disk of the PC. The method is sent to PalmSens again by using the command ‘L’.

1.2.2 Receiving data from PalmSens
The format of the information sent with ‘U’ and ‘T’ is:

‘U’ is followed by 12 characters of ‘0’ --- ‘9’, ‘A’---‘F’. Each two characters gives an integer in the range 0 to 255.
The first four characters represent the potential: intLE, intHE with
\[ E = \frac{(\text{intHE} \times 256 + \text{intLE})}{65536} \times 4.096 - 2.048 \]
The next four characters represent the current: intLI, intHI with
\[ I = \frac{(\text{intHI} \times 256 + \text{intLI})}{65536} \times 4.096 - 2.048 \times \text{current range}. \]
The next two characters are reserved for other applications.
The next two characters give the status and the applied current range: intStatus with
Current range = \[ 10^{\text{intStatus and 0Fh}} \] in nA units.
Current overload is detected when (intStatus AND 010h) is 10h.
A higher current range is required for proper measurements.
Current underload is detected when intStatus AND 040h) is 40h.
A lower current range can be applied for better resolution.
Voltage overload when (intStatus AND 020h) is 20h. The applied potential is not correct.
This is due to a too high impedance between counter and working electrode.

‘T’ is followed by 20 characters. The first 12 characters are identical to the string of ‘U’. The additional information is:
Four characters specifying the approximate temperature in degrees Celcius:
\[ \text{temp} = 25 - \frac{1000}{3} \times \frac{\text{intLT} + \text{intHT} \times 256}{65536} \times 4.096 - 0.63 \]
An accurate temperature measurement requires calibration. Please note that the temperature of the microcontroller chip is given.
The last four characters specify the noise level: intLN, intHN
\[ \text{Noise} = \frac{(\text{intLN} + \text{intHN} \times 256)}{1000} \times \text{current range} \]
The noise value is given as a root-mean-square assumed that the noise shape is sinusoidal.

‘S’ is followed by four characters intNL, intNH specifying the number of data points (E, I and status) which is sent.
\[ \text{Npoints} = \text{intNH} \times 256 + \text{intNL} \]
These data points are subsequently received. Each data point is preceded by the character ‘U’.

‘V’ is followed by four characters intNL, intNH specifying the number of data points (I or E) which is sent.
\[ \text{Npoints} = \text{intNH} \times 256 + \text{intNL} \]
These data points are subsequently received. Each data point is preceded by the character ‘Y’.

‘Y’ is followed by a datapoint of a chronoamperometric curve.
The four characters represent the potential: intLI, intHI with
\[ i = \frac{(\text{intHI} \times 256 + \text{intLI})}{65536} \times 4.096 - 2.048 \times \text{current range} \]

Page 4 of 4
1.2.3 Some examples

Please note that the spaces are added for clarity.

T 4A9F 2D9F 00 03 2629 0100 yields :
  E = 0.501 V (4A9Fh)
  I = 0.4988 E -06 A (2D9Fh + 03h))
  I(noise) = 0.001 E-06 A (0100h)
  Status : no overloads or underload (03h)
  T = 15.5º C (2629h)

T C679 2641 00 02 1229 1B00 yields :
  E = -0.100 V (C679h)
  I = -1.0056 E-07 A (2641h + 02h))
  I(noise) = 0.030 E-07 A (1B00h)
  Status : no overloads or underload (03h)
  T = 16.0º C (1229h)

Note:
Communication routines written for Visual Basic are available upon request.
2. Remote control of PalmSens

PalmSens is remote controlled by sending the character ‘c’ followed by a character and two integers each of two bytes with a value from 0 (hexadecimal 00h) to 255 (hexadecimal FFh). These characters and two integers are sent immediately after ‘c’ has been received by the PC. Be sure that in the period between sending ‘c’ and sending the command, a character and two hexadecimal values, no other information is sent to PalmSens.

2.1 Fast remote control mode.
PalmSens normally performs continuously measurements of the potential, current and temperature. This means that remote control may have a response of up to 100 ms.
In case PalmSens has to be used in the fast remote control mode, the command ‘J’ has to be sent. If the command ‘j’ is sent, PalmSens returns to its normal remote control mode.
(Note: Available in firmware version 2.7 and higher)

2.2 Set potential or current
‘D’, intL, intH
Set E or I with setting intH/intL as high/low bytes respectively. The values intH and intL are calculated according to the algorithm given below. The range of Vapplied is –2.048 V to 2.047 V. The applied potential E is equal to Vapplied if the mode is potentiostatic.
In the mode the galvanostatic the applied current I is equal to (Vapplied * current range)

\[
\text{intN} = (V_{\text{applied}} + 2.048) \times 1000
\]

\[
\text{intH} = \text{Int}(\text{intN} / 256) \quad \text{‘calculate high byte as integer}
\]

\[
\text{intL} = \text{Int}(\text{intN} - 256 \times \text{intH}) \quad \text{‘calculate low byte as integer}
\]

Example: Send ‘c’, wait until ‘c’ has been received back and then send ‘D0008’.
Set E to 0.000 V or I to 0.000 * current range.

2.2.1 Set potential of WE2
‘d’, intL, intH
The range for E(WE2) is –2.048 V to + 2.047 V, with a resolution of 1 mV. The resolution of the DA converter is therefore 12 bits. Please note the DA converter used to set the potential of WE2 is the same as analog output available on the mini-DIN connector.

\[
\text{intN} = (E(\text{WE2}) + 2.048) \times 1000
\]

\[
\text{intH} = \text{Int}(\text{intN} / 256)
\]

\[
\text{intL} = \text{Int}(\text{intN} - 256 \times \text{intH})
\]

Example: Send ‘c’, wait until ‘c’ has been received back and then send ‘d0008’.
Set external output voltage to 0.000 V. See note about the offset below.
2.2.2 Set analog output on miniDIN connector.
'd', intL, intH
Set external analog output with setting intH/intL. The values intH and intL are calculated according to the algorithm given below. The range for Vanalog is 0.000 V to + 4.095 V, with a resolution of 1 mV. The resolution of the DA converter is therefore 12 bits.

\[
\text{intInt} = \text{Vanalog} \times 1000 \\
\text{intH} = \text{Int(}\text{intInt} / 256) \\
\text{intL} = \text{Int(}\text{intInt} - 256 \times \text{intH})
\]

Example: Send 'c', wait until 'c' has been received back and then send 'd0008'.
Set external output voltage to 2.048 V.

2.2.3 Set mode of potentiostat/galvanostat and current range.
'G', intCR, intMode
Set mode of potentiostat/galvanostat and current range.
The bits of intMode are:
- d0 0: galvanostatic mode  1: potentiostatic mode
- d1 0: cell off   1: cell on
- d2 0: CA feedback off  1: CA feedback on
- d3-d5 stability setting used in galvanostatic mode
  000: if current range is 1 µA
  001: if current range is 10 µA
  010: if current range is 100 µA
  100: if current range is 1 mA
- 100: if potentiostatic mode is used
- d6-d7 nor used

Examples: intMode = 0Bh potentiostat, cell on, CA feedback off
          intMode = 0Dh potentiostat, cell off, CA feedback on
          intMode = 22h galvanostat, cell on, CA feedback off
                     stability setting for current range 10 µA

The current range is specified by intCR.
- intCR=0     current range is 1 nA
- intCR=1     current range is 10 nA
- intCR=2     current range is 100 nA
- intCR=3     current range is 1 µA
- intCR=4     current range is 10 µA
- intCR=5     current range is 100 µA
- intCR=6     current range is 1 mA

The actual current range therefore is 10 ^ intCR in units of nA.

Example: Send 'c', wait until 'c' has been received back and then send 'G030B'.
Set potentiostat, cell on, current range 1 µA.
2.2.4 Set BiPotentiostat module

\texttt{W}, \texttt{intCR}, \texttt{intDummy}

Set current range of the bipotentiostat module and switch WE2 on:

\begin{itemize}
  \item \texttt{intCR=1} or hex 01 current range is 1 nA
  \item \texttt{intCR=2} or hex 02 current range is 10 nA
  \item \texttt{intCR=4} or hex 04 current range is 100 nA
  \item \texttt{intCR=8} or hex 08 current range is 1 µA
  \item \texttt{intCR=16} or hex 10 current range is 10 µA
  \item \texttt{intCR=32} or hex 20 current range is 100 µA
  \item \texttt{intCR=64} or hex 40 current range is 1 mA
  \item \texttt{intCR=128} or hex 80 current range is 10 mA
\end{itemize}

Switch WE 2 off:

\texttt{intCR= 188} or hex BC

Example: Send 'c', wait until ‘c’ has been received back and then send ‘W0800’.
Switch WE2 on and set current range 1 µA.

The bipotentiostat module gives information about the overload or underload status of the current measurement.
The status of the overload- and underload-bits of the current measurement is reported when the command \texttt{r}, \texttt{intDummy}, \texttt{intDummy} is sent to PalmSens (the values of the two controlbytes which have to be sent with “r” are not relevant).
PalmSens responds with a string “rxxxH” where the x character is not relevant.
The value of “H” gives the status of the current follower:
\begin{itemize}
  \item 6 = normal
  \item 4 = current overload, which means that the current is higher than 1.6 * current range
  \item 2 = underload, which means that the current is below 0.05 * current range
\end{itemize}

2.3 Reading the current of WE2

The current of WE2 is measured with ADC channel 2. The current is recorded when ‘a’ command is used.

Example: Send 'c', wait until ‘c’ has been received back and then send ‘a0200’.
PalmSens sends the result of the current measurement as described below in the paragraph ‘Read analog input signal’

The current is calculated using:
\begin{equation}
  \text{Curr} = (V_{adc} - 2.048 - \text{loffset}) \times \text{current range}
\end{equation}

The value of loffset is determined by the procedure described in par. 2.4.

2.4 Offset of BiPotentiostat:

The bipotentiostat offsets have to be determined.

The measurement of the current of the bipotentiostat (loffset) by the ADC of PalmSens has an offset. This value is measured after sending 124 or hex 7C to the BiPot.
Correct all the current reading (in V) by subtracting the offset value.
Normal offset values are around: \text{loffset} = 0.010 – 0.020.

The potential E(WE2) of WE2 also has an offset.
Measurement of the potential offset is done by sending 58 or hex 3A to the bipotentiostat using the command ‘W’.
First set the potential of WE2 to 0 V. Switch off the cell and connect RE to AGND.
Potential offset equals: E2offset = - (I2out-loffset) / 100
Correct the potential E2 as set, by subtracting E2Offset from the value.
Normal offset values are around -0.004 to -0.020
2.5 Read current, potential or auxiliary analog input signal.

'a', ichannel, intDummy
Read analog input signal on channel ichannel. The value ichannel is in the range 0 to 7. The channel on the miniDIN connector is selected with ichannel = 5.
The current is read at channel 0 and the potential at channel 1.
The value which is returned is an average of 1024 conversions done in 20 ms (50 Hz) or 16.7 ms (60 Hz).
The result of the AD conversion is returned as 'a', intH, intL. The analog input signal equals:
Vin= (intH*256 + intL) / 65536 * 4.096
If channel 0 is used, the current value is: (Vin-2.048) * Current Range.
The potential read from channel 1 is: Vin -2.048 V.

Please note that the result is given with 16 bits, however the AD converter has an actual resolution of 12 bits, which gives a resolution of 1 mV.
Example: Send 'c', wait until 'c' has been received back and then send 'a05FF'.
Read the voltage at the auxiliary input on the miniDIN connector.

2.6 Read current, potential or auxiliary analog input signal continuously at fixed rate

'b', ichannel, intTime
Read analog input signal on channel ichannel. The value ichannel is in the range 0 to 7. The channel on the miniDIN connector is selected with ichannel = 5.
The current is read at channel 0 and the potential at channel 1.
The channel on the miniDIN connector is 5.
The value which is returned is an average of 1024 conversions done in 20 ms or 16.7 ms.

The interval time between two samples is equal to:
intTime * 2.21 ms + 21 ms, if the sampling period is 20 ms (at 50 Hz) or
intTime * 2.21 ms + 17.7 ms if the sampling time is 16.7 (at 60 Hz).
The sampling continues until PalmSens receives any other command is sent to PalmSens
The result of the AD conversion is returned as 'b', intH, intL. The analog input signal equals:
Vin= (intH*256 + intL) / 65536 * 4.096
If channel 0 is used, the current value is: (Vin-2.048) * Current Range.
The potential read from channel 1 is: Vin -2.048 V.

Please note that the result is given with 16 bits, however the AD converter has an actual resolution of 12 bits, which gives a resolution of 1 mV.
Example: Send 'c', wait until 'c' has been received back and then send 'a05FF'.
Read the voltage at the auxiliary input on the miniDIN connector.

(Note: Available in firmware version 2.8 and higher)

2.7 Set digital output lines of miniDIN connector.

'v', intOut, intDummy
Set external digital output lines with setting intOut. Only d0, d1 and d2 are relevant.
The range of intOut therefore is 0 to 7. The value of intDummy is not relevant and can have any value from 0 to 255.
Example: Send 'c', wait until 'c' has been received back and then send 'v07FF'.
Set all three digital output lines to 5 V.
2.8 Read digital input line of miniDIN connector.
'r', intDummy,intDummy
Read the digital input ports. The result of this command is returned:
'r', intExt, intInt
The intExt value gives the status of the input line on the miniDIN connector. Its value is either 0 or 1, so only
d0 is relevant.
The value of intInt is obtained from an internal digital input port available on the internal bus. In the standard
configuration this value is not relevant.
Example: Send 'c', wait until 'c' has been received back and then send 'rFFFF'.
Read digital input line.

2.9 Sending and receiving integer values as ASCII characters.
All communication is done by means of ASCII characters. Conversion of two ASCII characters to a decimal
value in the range 0 – 255 is done as follows:
Asca = Asc(Mid(strHex, 2, 1)) 'get ASCII number of second character
If Asca > 47 And Asca < 58 Then intInt = Asca - 48
If Asca > 64 And Asca < 71 Then intInt = Asca - 55
Asca = Asc(Mid(strHex, 1, 1)) 'get ASCII number of first character
If Asca > 47 And Asca < 58 Then intInt = intInt + 16 * (Asca - 48)
If Asca > 64 And Asca < 71 Then intInt = intInt + 16 * (Asca - 55)
The range of intInt is 0 to 255.

2.9.1 Sending an integer to PalmSens in the range of 0 to 255 is done as follows:
'intInt is the value to be sent.
strTemp = Hex(intInt) 'convert hexadecimal value to string of two characters.
If Len(strTemp) > 0 Then
    If Len(strTemp) = 1 Then strTemp = "0" + strTemp
    If Len(strTemp) > 2 Then strTemp = Right(strTemp, 2)
    strCommOut = Mid(strTemp, 1, 1)
    Call SendChar(strCommOut)
    strCommOut = Mid(strTemp, 2, 1)
    Call SendChar(strCommOut)
End If

2.10 Receiving PalmSens versions and power status
't'
PalmSens returns the hardware and software version and it’s power status when it receives the character
't'. The result of this command is returned: 'palmXXZZ', where:
XX=Hardware version and power status
ZZ=Firmware version
If (XX AND 01) = 0 then PalmSens is charging the batteries
If (XX AND 02) = 0 then the batteries are low
If (XX AND 08) = 0 then PalmSens version 1 is used
If (XX AND 08) = 1 then PalmSens version 2 is used, with an additional 10 mA current range
ZZ represents the firmware version without the decimal point, e.g. when ZZ = 30, the firmware version is
3.0.
3. Method parameters

The method parameters are sent to PalmSens by using the command ‘L’. These parameters can be received from PalmSens using the command ‘K’.
Most of the parameters are specified by two bytes. These values range from 0 – 65535.

Some parameters can be specified by the user. All parameters are given in an array with Par(0) as first element and Par(65) as last element.

Potential values:
- \( E_{begin} : \) \( \text{Par}(16) + \text{Par}(17) \) (resp. ByteLow + ByteHigh)
- \( E_{pulse} : \) \( \text{Par}(18) + \text{Par}(19) \)
- \( E_{step} : \) \( \text{Par}(20) + \text{Par}(21) \)
- \( E_{by} : \) \( \text{Par}(26) + \text{Par}(27) \)
- \( E_{cond} : \) \( \text{Par}(28) + \text{Par}(29) \)
- \( E_{dep} : \) \( \text{Par}(30) + \text{Par}(31) \)
- Technique : \( \text{Par}(33) \)
- Number of steps: \( \text{Par}(34) + \text{Par}(35) \)

Time values:
- \( T_{cond} : \) \( \text{Par}(10) + \text{Par}(11) \) (resp. ByteLow + ByteHigh)
- \( T_{dep} : \) \( \text{Par}(12) + \text{Par}(13) \)
- \( T_{eq} : \) \( \text{Par}(14) + \text{Par}(15) \)

3.1 Calculation of potential values :
In the next example \( E \) can be \( E_{begin} \), \( E_{pulse} \), \( E_{step} \), \( E_{cond} \) and \( E_{dep} \)

\[
\text{IntValue} = \text{Int} ((E + 2.048) * 1000)
\]
\[
\text{ByteHigh} = \text{Int} (\text{IntValue} / 256)
\]
\[
\text{ByteLow} = \text{IntValue} - 256 * \text{ByteHigh}
\]

Note:
\( E_{end} \) is not a PalmSens parameter. It can be calculated from \( E_{begin} \), \( E_{step} \) and the number of steps: \( \text{End} = E_{begin} + (\text{IntNsteps}-1) * E_{step} \).

Technique:
The number used are :
- 0: linear sweep voltammetry
- 1: differential pulse voltammetry
- 2: square wave voltammetry
- 3: normal pulse
- 4: ac-voltammetry
- 5: cyclic voltammetry
- 6: PSA
- 7: amperometric detection
- 8: pulsed amperometric detection
- 9: fast chrono-amperometry

Page 12 of 12
**Number of steps:**
\[ \text{IntNsteps} = \text{ABS}(\text{Int}((\text{Eend} - \text{Ebegin})/\text{Estep})) + 1 \]

**Calculation of time values:**
In the next example T can be Tcond, Tdep and Teq).
\[ \text{IntValue} = \text{Int} (T \times 16) \]
\[ \text{ByteHigh} = \text{Int} (\text{IntValue} / 256) \]
\[ \text{ByteLow} = \text{IntValue} - 256 \times \text{ByteHigh} \]

**Current ranges:**
In the next example CR is the initial, lowest or highest current range, specified in A.
\[ \text{IntCR} = \text{Int} (\log ((\text{CR} \times 1E9) / \log (10) + 0.5)) \]
Appendix A: Manual control flow chart

'c' to PalmSens

'c' received on PS
Send 'c' back
Wait for command

'c' received on PC
Send command (format: 'cmd, int, int')

Command received on PS
Carry out command

Is a value to be returned to PC?

Value calculated on PS
Send value to PC (format: 'cmd, int, int')
End

no

yes
Appendix B

Detailed Circuitry Schematic
Appendix C

LabView Code (Printable Version)
<table>
<thead>
<tr>
<th>Voltage</th>
<th>Temperature (degC)</th>
<th>Hardware</th>
<th>Firmware</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.000</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Current (nA)</td>
<td>I Range (nA)</td>
<td>Noise</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

**Power Status**

**Batt Status**

**COM Port**

- COM8
Pocket PC Portrait Screen

Use this template to create a PDA VI.

Front Panel
Evaluation

Pocket PC Portrait Screen

G:\Personal\LV for Laptop\LabVIEW Data\Bluetooth.vi

Last modified on 4/10/2008 at 12:02 AM
Printed on 11/5/2008 at 6:40 PM

Type

[0] Timeout

Source

[1] "LOAD PARAMETERS": Value Chan

[1] "LOAD PARAMETERS": Value Chan
status iteration
gets temp, etc from PS and runs its last iteration after a button is pressed that is cause for delay on other screens
Evaluation
Pocket PC Portrait Screen
G:\Personal\LV for Laptop\LabVIEW Data\Bluetooth.vi
Last modified on 4/10/2008 at 12:02 AM
Printed on 11/5/2008 at 6:40 PM

[2] "MEASURE!": Value Change

Source

Boolean

[3] "SET VALVE PARAMETERS 2": Value Change

Source
Type
Time
CtlRef
OldVal
NewVal

Boolean

[0] Timeout

Type
[1] "MEASURE!": Value Change

Source

Boolean

M command (SubVI).vi
List of SubVIs and Express VIs

**Pump and Valve Controls Debug Fluidic.vi**
G:\Personal\LV for Laptop\LabVIEW Data\Pump and Valve Controls Debug Fluidic.vi

**Release Semaphore.vi**
C:\Program Files\National Instruments\LabVIEW 8.6\vi.lib\Utility\semaphor.lib\Release Semaphore.vi
Connector Pane

time delay global var.vi

Front Panel

Numeric

Block Diagram

List of SubVIs and Express VIs
T command (SubVI).vi
Palmsens is charging the batteries

Battery power low

Current overload

Current Underload

Voltage Overload

VISA resource

Error out
**Voltage (SubVI).vi**
G:\Personal\LV for Laptop\LabVIEW Data\Voltage (SubVI).vi

**Noise (SubVI).vi**
G:\Personal\LV for Laptop\LabVIEW Data\Noise (SubVI).vi
<table>
<thead>
<tr>
<th>Technique</th>
<th>Power Line</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear Sweep Voltammetry</td>
<td>60Hz</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Ebegin (V)</th>
<th>Eend (V)</th>
<th>Eamplitude (V)</th>
<th>Estep (V)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Econd (V)</th>
<th>Edep (V)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.000</td>
<td>0.000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Tcondition</th>
<th>Tdeposition</th>
<th>Tequilibrium</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Highest I</th>
<th>Initial I</th>
<th>Lowest I</th>
</tr>
</thead>
<tbody>
<tr>
<td>1nA</td>
<td>1nA</td>
<td>1nA</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Frequency (Hz)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
</tr>
</tbody>
</table>
**Linear Sweep Voltammetry**

- **Ebegin (V)**: 0.000
- **Eend (V)**: 0.000
- **Emplitude (V)**: 0.000
- **Estep (V)**: 0.000
- **Econd (V)**: 0.000
- **Edep (V)**: 0.000
- **Tcondition**: 0.00
- **Tdeposition**: 0.00
- **Tequilibrum**: 0.00
- **Highest I**: 1nA
- **Initial I**: 1nA
- **Lowest I**: 1nA
- **Power Line**: 60Hz

**Technique**

- **Linear Sweep Voltammetry**

**CHANGE PARAMETERS**

**MAIN MENU**
time delay global var.vi
G:\Personal\LV for Laptop\LabVIEW Data\time delay global var.vi

10^(x-9) good (SubVI).vi
G:\Personal\LV for Laptop\LabVIEW Data\10^(x-9) good (SubVI).vi

L Command (SubVI).vi
G:\Personal\LV for Laptop\LabVIEW Data\L Command (SubVI).vi

HI x 256 + LO DIV 1000 good (SubVI).vi
G:\Personal\LV for Laptop\LabVIEW Data\HI x 256 + LO DIV 1000 good (SubVI).vi

Simple HEX conversion (SubVI).vi
G:\Personal\LV for Laptop\LabVIEW Data\Simple HEX conversion (SubVI).vi

HI x 256 + LO DIV 1000 -2.048 (SubVI).vi
G:\Personal\LV for Laptop\LabVIEW Data\HI x 256 + LO DIV 1000 -2.048 (SubVI).vi

HI x 256 + LO DIV 16 (SubVI).vi
G:\Personal\LV for Laptop\LabVIEW Data\HI x 256 + LO DIV 16 (SubVI).vi
Evaluation

10^(x-9) good (SubVI).vi

G:\Personal\LV for Laptop\LabVIEW Data\10^(x-9) good (SubVI).vi
Last modified on 2/22/2008 at 9:00 PM
Printed on 11/5/2008 at 6:46 PM

10^(x-9) good (SubVI).vi
HI x 256 + LO DIV 1000 -2.048 (SubVI).vi

G:\Personal\LV for Laptop\LabVIEW Data\HI x 256 + LO DIV 1000 -2.048 (SubVI).vi

Last modified on 2/22/2008 at 9:00 PM
Printed on 11/5/2008 at 6:46 PM
HI x 256 + LO DIV 1000 good (SubVI).vi

G:\Personal\LV for Laptop\LabVIEW Data\HI x 256 + LO DIV 1000 good (SubVI).vi

Last modified on 2/22/2008 at 9:00 PM
Printed on 11/5/2008 at 6:46 PM
Simple HEX conversion (SubVI).vi

Numeric

0.00

HE

LE

Numeric

0

256

HE

LE

1.23

0.00

LabVIEW™ Evaluation Software
**Square Wave Voltammetry**

- **Ebegin (V)**: -1.200
- **Eend (V)**: -0.400
- **Estep (V)**: 0.005
- **Epulse (V)**: 0.001

- **Econd (V)**: -1.000
- **Edep (V)**: -1.200

- **Tcondition**: 0
- **Tdeposition**: 20
- **Tequilibrium**: 0

- **Highest I**: 1mA
- **Initial I**: 10uA
- **Lowest I**: 10nA
- **Scan Rate (V/s)**: 0.000

**SWV parameter only**

- **Frequency (Hz)**: 100.00
- **Eamplitude (V)**: 0.025

**Cyclic Voltammetry Only**

- **Evertex1**: 0.000
- **Evertex2**: 0.000
- **Number of Scans**: 1
[2] "SET PARAMETERS": Value Change
Basic L Conversion (SubVI).vi

Diagram showing conversion between numeric and string data types with inputs and outputs labeled as follows:
- Numeric inputs: 4, 256, 2, etc.
- String outputs: "abc"
Global 1.vi

Boolean
**I1-I2 Globals.vi**

- **I1-I2 generic**: 0
- **V generic**: 0
- **V 2**: 0
- **V 3**: 0
- **V 1**: 0

**I1-I2 val1**: 0

**I1-I2 val2**: 0

**I1-I2 val3**: 0
Simple HEX conversion (SubVI).vi
time delay global var.vi

Numeric

0
Voltage for L (SubVI).vi
Max Current (uA)
8842.67481
Voltage at Max Current
0.00

Y = mx + b slope

Max I Minus Base I (uA)
8842.67481
Sample Duty Cycle
Sample/sec
Freq
Cont Meas?
True
Offset
Ampl
Sample Duty Cycle
VISA resource name out 2
error out 2
Numeric
DBL
Evaluation

M command trial(SubVI).vi

G:\Personal\LV for Laptop\LabVIEW Data\M command trial(SubVI).vi

Last modified on 4/24/2008 at 5:22 PM

Printed on 11/5/2008 at 6:58 PM

Wait 20s for user RHS Limit

20000
file path global.vi
G:\Personal\LV for Laptop\LabVIEW Data\file path global.vi
M command (SubVI).vi

VISA resource name out
VISA resource name out 2
Numeric
error out
error out 2

Plot 0
I-V Characteristics

Max Current
0.00
Voltage at Max Current
0.00

BACK
read junk

return count

500
Write To Spreadsheet File (DBL).vi
C:\Program Files\National Instruments\LabVIEW 8.6\vi.lib\Utility\file.lib\Write To Spreadsheet File (DBL).vi

Write To Spreadsheet File.vi
C:\Program Files\National Instruments\LabVIEW 8.6\vi.lib\Utility\file.lib\Write To Spreadsheet File.vi

Voltage (SubVI).vi
G:\Personal\LV for Laptop\LabVIEW Data\Voltage (SubVI).vi
### Pump and Valve Controls Debug Fluidic.vi

<table>
<thead>
<tr>
<th>Step</th>
<th>Sample Number</th>
<th>Delay before Measurement (sec)</th>
<th>Delay before Step (sec)</th>
<th>Sensor</th>
<th>Sensor 1</th>
<th>Sensor 2</th>
<th>Sensor 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Output Data Files are saved**

**Buffer Up/Dn**

**Pump Continuously during Predep?**
Source

START

Pump Continuously during Predep?

Auto (ON)/ Manual (OFF)

Buffer Pump Time (sec)

Sample Pump Time (sec)

Ampl

Freq

Offset

Sample Duty Cycle

Buffer Flow rate (uL/min)

Sample Flow rate (uL/min)

Send Pump SW on AO0

Sample Flow rate (uL/min)

Auto/manual

Cont Meas?
Which valve?
Valve initialization done - start control

Choose which valves are opened and set the switches for direction;
Sensor 1

Sensor Address (SubVI).vi

Sensor 1

Connect Sensor

Delay

0 1000

Boolean

turn pump on and set which sensor is used
Sensor 2

VISA resource name 2

error in (no error) 2

Sensor Address (Sensor 2)
Measure with loaded parameters

Path current

Path Voltage

I1-I2 generic

V 3

V generic

I1-I2 val3

True
linked to semaphore
Square wave generation to AO for pump.vi
G:\Personal\LV for Laptop\LabVIEW Data\Square wave generation to AO for pump.vi

DAQmx Clear Task.vi
C:\Program Files\National Instruments\LabVIEW 8.6\vi.lib\DAQmx\configure\task.llb\DAQmx Clear Task.vi

DAQmx Write.vi
C:\Program Files\National Instruments\LabVIEW 8.6\vi.lib\DAQmx\write.llb\DAQmx Write.vi

DAQmx Write (Digital Bool 1Line 1Point).vi
C:\Program Files\National Instruments\LabVIEW 8.6\vi.lib\DAQmx\write.llb\DAQmx Write (Digital Bool 1Line 1Point).vi

M command trial(SubVI).vi
G:\Personal\LV for Laptop\LabVIEW Data\M command trial(SubVI).vi

file path global.vi
G:\Personal\LV for Laptop\LabVIEW Data\file path global.vi

Pump Globals.vi
G:\Personal\LV for Laptop\LabVIEW Data\Pump Globals.vi

I1-I2 Globals.vi
G:\Personal\LV for Laptop\LabVIEW Data\I1-I2 Globals.vi

Average 3 sensors.vi
G:\Personal\LV for Laptop\LabVIEW Data\Average 3 sensors.vi

Pumping boolean global.vi
G:\Personal\LV for Laptop\LabVIEW Data\Pumping boolean global.vi

time delay global var.vi
G:\Personal\LV for Laptop\LabVIEW Data\time delay global var.vi

Switch position (SubVI).vi
G:\Personal\LV for Laptop\LabVIEW Data\Switch position (SubVI).vi

Valve Address (SubVI).vi
G:\Personal\LV for Laptop\LabVIEW Data\Valve Address (SubVI).vi

Sensor Address (SubVI).vi
G:\Personal\LV for Laptop\LabVIEW Data\Sensor Address (SubVI).vi

pulse generation on AO1(alt) (SubVI).vi
G:\Personal\LV for Laptop\LabVIEW Data\pulse generation on AO1(alt) (SubVI).vi
32 bit adder.vi
**Average 3 sensors.vi**

- C1-3
- C1-2
- C1-1

**Average File**
- V1-1
- V1-2
- V1-3
- S1
- S2
- S3

**Average Peak Current (uA)**

**Voltage at Average Peak**

**Max Current (uA)**

**Voltage at Max Current**

**I-V Characteristics 1F**

*Plot 0*

![Graph showing I-V characteristics with voltage (V) on the y-axis and current (A) on the x-axis.*

**National Instruments**
**pulse generation on AO1(alt) (SubVI).vi**

milliseconds to wait

---

**DAQmx Create Virtual Channel.vi**
C:\Program Files\National Instruments\LabVIEW 8.6\vi.lib\DAQmx\create\channels.llb\DAQmx Create Virtual Channel.vi

**DAQmx Start Task.vi**
C:\Program Files\National Instruments\LabVIEW 8.6\vi.lib\DAQmx\configure\task.llb\DAQmx Start Task.vi

**DAQmx Write.vi**
C:\Program Files\National Instruments\LabVIEW 8.6\vi.lib\DAQmx\write.llb\DAQmx Write.vi

**DAQmx Clear Task.vi**
C:\Program Files\National Instruments\LabVIEW 8.6\vi.lib\DAQmx\configure\task.llb\DAQmx Clear Task.vi

**DAQmx Write (Analog DBL 1Chan 1Samp).vi**
C:\Program Files\National Instruments\LabVIEW 8.6\vi.lib\DAQmx\write.llb\DAQmx Write (Analog DBL 1Chan 1Samp).vi

**DAQmx Create Channel (AO-Voltage-Basic).vi**
C:\Program Files\National Instruments\LabVIEW 8.6\vi.lib\DAQmx\create\channels.llb\DAQmx Create Channel (AO-Voltage-Basic).vi
**pulse generation on p0.0(SubVI).vi**

- milliseconds to wait
- error in
- error out

- Digital Output
- one channel for each line

**Examples:**
- **DAQmx Create Virtual Channel.vi**
  C:\Program Files\National Instruments\LabVIEW 8.6\vi.lib\DAQmx\create\channels.llb\DAQmx Create Virtual Channel.vi
- **DAQmx Create Channel (DO-Digital Output).vi**
  C:\Program Files\National Instruments\LabVIEW 8.6\vi.lib\DAQmx\create\channels.llb\DAQmx Create Channel (DO-Digital Output).vi
- **DAQmx Start Task.vi**
  C:\Program Files\National Instruments\LabVIEW 8.6\vi.lib\DAQmx\configure\task.llb\DAQmx Start Task.vi
- **DAQmx Write.vi**
  C:\Program Files\National Instruments\LabVIEW 8.6\vi.lib\DAQmx\write.llb\DAQmx Write.vi
- **DAQmx Write (Digital Bool 1Line 1Point).vi**
  C:\Program Files\National Instruments\LabVIEW 8.6\vi.lib\DAQmx\write.llb\DAQmx Write (Digital Bool 1Line 1Point).vi
- **DAQmx Clear Task.vi**
  C:\Program Files\National Instruments\LabVIEW 8.6\vi.lib\DAQmx\configure\task.llb\DAQmx Clear Task.vi
Pumping boolean global.vi

STOP

Boolean

Boolean 1

Boolean 2

Boolean 3

Boolean 4

Boolean 5

Boolean 6

Boolean 7
**Pump Globals.vi**

- **Freq**: 0
- **Ampl**: 0
- **Cont Meas?**
- **Sample Duty Cycle**: 0
- **Sample/sec**: 0
- **Offset**: 0
Digital Bool
1Line 1Point

one channel for each line

Digital Output
Address which sensor is turned on to relay to P element3

- Dev1/port0/line6
  - one channel for each line

Digital Output

Digital Bool 1Line 1Point

one channel for each line
Square wave generation to AO for pump.vi

- SW Freq
- Samples/sec
- Duty cycle (%)
- Amplitude
- Offset voltage

DAQmx Clear Task.vi
C:\Program Files\National Instruments\LabVIEW 8.6\vi.lib\DAQmx\configure\task.llb\DAQmx Clear Task.vi
Evaluation

Square wave generation to AO for pump.vi

G:\Personal\LV for Laptop\LabVIEW Data\Square wave generation to AO for pump.vi

Last modified on 2/22/2008 at 9:00 PM

Printed on 11/5/2008 at 7:03 PM

- **DAQmx Write (Analog DBL 1Chan 1Samp).vi**
  C:\Program Files\National Instruments\LabVIEW 8.6\vi.lib\DAQmx\write.llb\DAQmx Write (Analog DBL 1Chan 1Samp).vi

- **DAQmx Write.vi**
  C:\Program Files\National Instruments\LabVIEW 8.6\vi.lib\DAQmx\write.llb\DAQmx Write.vi

- **DAQmx Start Task.vi**
  C:\Program Files\National Instruments\LabVIEW 8.6\vi.lib\DAQmx\configure\task.llb\DAQmx Start Task.vi

- **Simple Error Handler.vi**
  C:\Program Files\National Instruments\LabVIEW 8.6\vi.lib\Utility\error.llb\Simple Error Handler.vi

- **DAQmx Create Channel (AO-Voltage-Basic).vi**
  C:\Program Files\National Instruments\LabVIEW 8.6\vi.lib\DAQmx\create\channels.llb\DAQmx Create Channel (AO-Voltage-Basic).vi

- **DAQmx Create Virtual Channel.vi**
  C:\Program Files\National Instruments\LabVIEW 8.6\vi.lib\DAQmx\create\channels.llb\DAQmx Create Virtual Channel.vi

- **NI_MABase.lvlib:Basic Function Generator.vi**
  C:\Program Files\National Instruments\LabVIEW 8.6\vi.lib\Measure\masignal.llb\Basic Function Generator.vi
Switch position (SubVI).vi

Boolean Switch position (SubVI).vi

Digital Output
one channel for each line

DAQmx Create Virtual Channel.vi
C:\Program Files\National Instruments\LabVIEW 8.6\vi.lib\DAQmx\create\channels.llb\DAQmx Create Virtual Channel.vi

DAQmx Create Channel (DO-Digital Output).vi
C:\Program Files\National Instruments\LabVIEW 8.6\vi.lib\DAQmx\create\channels.llb\DAQmx Create Channel (DO-Digital Output).vi

DAQmx Start Task.vi
C:\Program Files\National Instruments\LabVIEW 8.6\vi.lib\DAQmx\configure\task.llb\DAQmx Start Task.vi

DAQmx Clear Task.vi
C:\Program Files\National Instruments\LabVIEW 8.6\vi.lib\DAQmx\configure\task.llb\DAQmx Clear Task.vi

DAQmx Write.vi
C:\Program Files\National Instruments\LabVIEW 8.6\vi.lib\DAQmx\write.llb\DAQmx Write.vi

DAQmx Write (Digital Bool 1Line 1Point).vi
C:\Program Files\National Instruments\LabVIEW 8.6\vi.lib\DAQmx\write.llb\DAQmx Write (Digital Bool 1Line 1Point).vi
### time parameters.vi

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Par 65 New</td>
<td>0</td>
</tr>
<tr>
<td>Par 66</td>
<td>0</td>
</tr>
<tr>
<td>Tmeas orig</td>
<td>0</td>
</tr>
<tr>
<td>TmeasAtAD</td>
<td>0</td>
</tr>
<tr>
<td>Par 65</td>
<td>0</td>
</tr>
<tr>
<td>Par 66</td>
<td>0</td>
</tr>
<tr>
<td>Toverhead</td>
<td>0</td>
</tr>
<tr>
<td>DT16ad</td>
<td>0</td>
</tr>
<tr>
<td>iADDelay new</td>
<td>0</td>
</tr>
<tr>
<td>int AD Cycles</td>
<td>0</td>
</tr>
<tr>
<td>intAD new</td>
<td>0</td>
</tr>
<tr>
<td>real Tmeas</td>
<td>0</td>
</tr>
</tbody>
</table>

![LabVIEW screenshot](attachment:image.png)
Toverhead 2

0.003

Par 65 New 2

66

ADDelay new 2

ADT16ad 2

orig 2
Valve Address (SubVI).vi

Valve Number

Digital Output

one channel for all lines

Digital U8 1Chan 1Samp

500

DAQmx Create Virtual Channel.vi
C:\Program Files\National Instruments\LabVIEW 8.6\vi.lib\DAQmx\create\channels.llb\DAQmx Create Virtual Channel.vi

DAQmx Create Channel (DO-Digital Output).vi
C:\Program Files\National Instruments\LabVIEW 8.6\vi.lib\DAQmx\create\channels.llb\DAQmx Create Channel (DO-Digital Output).vi

DAQmx Start Task.vi
C:\Program Files\National Instruments\LabVIEW 8.6\vi.lib\DAQmx\configure\task.llb\DAQmx Start Task.vi

DAQmx Write.vi
C:\Program Files\National Instruments\LabVIEW 8.6\vi.lib\DAQmx\write.llb\DAQmx Write.vi

DAQmx Write (Digital U8 1Chan 1Samp).vi
C:\Program Files\National Instruments\LabVIEW 8.6\vi.lib\DAQmx\write.llb\DAQmx Write (Digital U8 1Chan 1Samp).vi

DAQmx Clear Task.vi
C:\Program Files\National Instruments\LabVIEW 8.6\vi.lib\DAQmx\configure\task.llb\DAQmx Clear Task.vi