A CMOS-based Lab-on-Chip Array for Combined Magnetic Manipulation and Opto-Chemical Sensing

Zheng Da Clinton Goh

Department of Bioengineering
Institute of Biomedical Engineering
Imperial College London

M.Eng in Biomedical Engineering
Final Year Project Report

Supervisor:
Dr. Pantelis Georgiou

Co-supervisors:
Dr. Timothy Constandinou,
Dr. Emm. Mic. Drakakis

7 June 2010

Abstract

A platform to integrate, control and process data from a CMOS-based lab-on-chip was developed for micro-scale bio-analysis. The lab-on-chip device is based on an 8×8 CMOS pixel array which integrates an ISFET chemical sensor, an active pixel sensor and an inductive micro-coil within each pixel. Fabricated in a 0.35μm CMOS technology, the lab-on-chip can be used for both optical detection and pH sensing and include mechanisms for calibrating out sensor mismatch. A spatiotemporal field pattern generator was also embedded within the array for magnetic micro-manipulation.

A test platform consisting of a microcontroller, external reference circuitry and MATLAB based user interfaces was designed to provide a user interface to the lab-on-chip. Experiments were controlled from a GUI which programs magnetic field patterns and performs real time data acquisition and sensor calibration through a microcontroller.

The lab-on-chip was characterised in terms of sensor sensitivity and drift, as well as actuation capabilities. The system successfully detected optical stimulus and demonstrated a maximum pH sensitivity of 57mV/pH with an exponentially decaying chemical drift. The system detected changes in electrolyte potential of 1.0mV and above and demonstrated significant improvement to signal-to-noise ratio after signal averaging across all pixels. Manipulation of paramagnetic micro-beads was also achieved by biasing a single pixel. The lab-on-chip was found to be pH insensitive when encapsulated with parylene, but detected micro-droplets of fluid and demonstrated sensitivity to electrolyte potential changes.

The novel CMOS-based lab-on-chip device can be used in various applications which utilise any combinations of the three sensing and actuation modalities. Through functional integration, new opportunities are created for full biochemical assays to be performed on a single micro-system.

Acknowledgements

I would like to express my heartfelt gratitude to Dr. Pantelis Georgiou for giving me the opportunity to take part in this project, and for his continuous guidance and support. I would also like to thank Dr. Timothy Constandinou for his precious advice and valuable insight, Dr. Themis Prodromakis for his patience and expertise and Liu Yan for his help in the laboratory. Thanks also goes to Dr. Emm. Drakakis, Dr. Nir Grossman, Virginia Woods, Panavy Pookaiyaudom and all other researchers and personnel both in the Institute of Biomedical Engineering and the Department of Bioengineering for their continuous support in this project.

Contents

1 Introduction **4**

2 Lab-on-Chip Devices **5**
2.1 Introduction . . . 5
2.2 Functional Building Blocks . . . 6
2.2.1 Chemical Sensors . . . 6
2.2.2 Optical Sensors . . . 7
2.2.3 Magnetic Actuators . . . 8
2.3 The T2P Project . . . 9

3 System Architecture **11**
3.1 Inductive Micro-Coil Arrays . . . 12
3.2 ISFET Arrays . . . 12
3.3 Photodiode Arrays . . . 13

4 Hardware Implementation **14**
4.1 Lab-on-Chip Development Board . . . 14
4.1.1 PIC Microcontroller . . . 15
4.1.2 Voltage References . . . 15
4.1.3 Inductive Micro-coil Current Sink . . . 15
4.1.4 ADC Clock . . . 15
4.2 Lab-on-Chip Cartridge . . . 16

5 Software Implementation **17**
5.1 Graphical User Interface Main Menu . . . 17
5.2 Magnetic Field Pattern Programmer . . . 17
5.3 Lab-on-Chip Initialisation . . . 18
5.4 Real Time Data Acquisition . . . 18
5.5 Image and Data Viewer . . . 19
5.6 Chemical Sensor Calibration . . . 20

6 System Verification **21**
6.1 Pointer Initialisation . . . 21
6.2 ADC Reference Voltage Initialisation . . . 22
6.3 Magnetic Field Pattern Programming . . . 22
6.3.1 3-Terminal Current Sink . . . 22
6.3.2 Magnetic Field Pattern Playback . . . 23
6.4 Real Time Data Acquisition . . . 24
6.4.1 Chemical Data Acquisition . . . 24
6.4.2 Optical Data Acquisition . . . 25
6.4.3 Real Time Data Acquisition with Magnetic Field Pattern Playback . . . 26
6.5 Chemical Sensor Calibration . . . 27

7 System Characterisation **29**
7.1 pH-Sensitivity . . . 29
7.2 Chemical Drift . . . 30
7.3 Electrolyte Potential Detection . . . 31
7.4 Magnetic Bead Movement . . . 33
7.5 Parylene Encapsulated Lab-on-Chip . . . 33
7.5.1 pH Sensitivity . . . 33
7.5.2 Electrolyte Potential Detection . . . 33

8 Conclusion **35**

9 Future Work **36**
9.1 Chemical Drift Compensation . . . 36
9.2 Noise Reduction . . . 36
9.3 Characterising Optical Sensitivity . . . 37
9.4 Characterising Crosstalk . . . 37
9.5 Characterising Magnetic Bead Movement . . . 37
9.6 Microfluidic Integration . . . 37
9.7 Application to Biochemistry and Cell Biology . . . 37

A Magnetic Force on a Super-paramagnetic Micro-bead **42**

B Operation of the PG-ISFET **43**

C Hardware Design Layout **44**

D MATLAB GUI Main Menu **47**

E Initialisation, Calibration and Real Time Data Acquisition GUI **48**

F Magnetic Field Pattern Simulation GUI **49**

G Image Analyser **50**

H Microcontroller Code Listing **51**
H.1 Main Program . . . 51
H.2 Microcontroller State Machine . . . 51
H.2.1 Global Variables . . . 51
H.2.2 Interrupts . . . 52
H.2.3 State Machine Main Function . . . 53
H.2.4 Program DACs . . . 54
H.2.5 Program Lab-on-Chip . . . 55
H.2.6 Initialise Lab-on-Chip . . . 56
H.2.7 Generate Magnetic Field Pattern . . . 56
H.2.8 Real Time Data Acquisition with Magnetic Field Pattern Generation . . . 57
H.2.9 Setup Magnetic RAM Playback Frequency . . . 57

I MATLAB Code Listings **59**
I.1 Serial Communication Setup . . . 59
I.2 MATLAB State Machine . . . 59
I.2.1 Program DACs . . . 59
I.2.2 Program Lab-on-Chip . . . 60
I.2.3 Read Sensor Data . . . 61
I.2.4 Display Sensor Data . . . 61
I.2.5 Initialise Lab-on-Chip . . . 62
I.2.6 Sensor Calibration . . . 62
I.2.7 Real Time Data Acquisition . . . 63
I.2.8 Magnetic Field Pattern Generation . . . 64
I.2.9 Real Time Data Acquisition with Magnetic Field Pattern Generation . . . 64

Chapter 1

Introduction

Current advancements in biomedical research are supported by new frontiers created from integrating complex functions on silicon technology. This combination of biochemistry, solid state physics, integrated circuit design and data processing has given birth to a multi-disciplinary field known as lab-on-chip.

Technology miniaturisation, embedded sensors and modern day micro-fabrication techniques are paving the way for the evolution of lab-on-chip technology towards highly complex and integrated systems. The need for size and cost reduction has prompted the use of CMOS technology to integrate low cost sensors, actuators and data processing circuits within a single system. With advancements in modern day CMOS technology, it is possible to replace traditional bulky and expensive equipment with micro-analytical systems of the future.

The design, testing and characterisation of a novel CMOS based lab-on-chip platform for combined magnetic manipulation and opto-chemical sensing is presented in this report. Simultaneous system monitoring with biological cell and molecule manipulation is made possible by integrating both sensing and actuation modalities within a single platform. As a result, this novel system may open up new opportunities in field of analytical biochemistry and cell biology.

This report first presents a review of current CMOS lab-on-chip devices, followed by a description of the system architecture, hardware and software implementation design for the user platform, system verification and system characterisation. The report is then concluded with suggestions for future work.

Chapter 2

Lab-on-Chip Devices

2.1 Introduction

The analysis of biological systems and miniaturisation of analytical chemical methods are few of the driving forces behind research in the field of Micro Total Analysis Systems (μTAS) or lab-on-chip. Lab-on-chip is a multi-disciplinary approach to designing micro-scale devices for performing biochemical assays and procedures. The advantages of technology miniaturisation are realised in microfluidic devices and systems which allow for controlled transport, immobilisation and manipulation of biological molecules and cells [1, 2, 3].

Integrating CMOS based microelectronics with lab-on-chip devices enable micro-scale spatiotemporal monitoring of microfluidic systems [4, 5, 6]. Typical CMOS based lab-on-chip devices use chemical or optical sensing modalities for monitoring biochemical assays, while others are designed for molecule and biological cell handling through magnetic actuation [7, 8, 9]. CMOS technology offers several advantages such as:

1. Integrated circuits: Modern day CMOS is experiencing a reduction in feature size as described by Moore's law [10, 11]. It is now possible to design very large-scale integrated (VLSI) circuits with complex multiplexing schemes, programmable actuators and embedded sensors [7].

2. Embedded sensing: Optical and chemical sensors can be easily integrated in sensor arrays with signal processing circuits for noise removal and feature detection [12, 13].

3. Embedded actuation: Cell manipulation can be achieved through electrostatic or electromagnetic actuators integrated within a CMOS chip [8].

4. Programmability: Array structured memory embedded in CMOS based lab-on-chip devices allow for greater controllability and reliability of biochemical assays through programmable sensors and actuators [7, 8].

5. Well established manufacturing technology: Microelectronic technology is now easily accessible and capable of producing large volumes of integrated circuit systems without significant investment in manufacturing facilities.

6. Structured design methodology: Readily available computer aided design (CAD) tools enable design automation and can be integrated into the design methodology for lab-on-chip devices.

Despite advances in CMOS based lab-on-chip technology, integration of multiple sensing and actuation modalities into a single programmable system has not been achieved. Previous designs of CMOS lab-on-chip devices focus mainly on systems which offer a single sensing or actuation modality [8, 13, 14]. These systems have demonstrated excellent performance capabilities, but only contribute to a small portion of a complete biochemical assay. Hence, the need for a general purpose system which can carry out a full biochemical protocol in real time and at low cost has driven efforts in functional integration.

2.2 Functional Building Blocks

2.2.1 Chemical Sensors

Chemical sensing can be achieved through floating gate devices such as the ion-sensitive field effect transistor (ISFET). ISFETs offer many advantages as they are cheap, low power, and provide a means for measuring samples in minute quantities. ISFET arrays can be used in various applications, such as:

1. Real time, in-vitro monitoring of bioelectrochemical activity of a living cell population. Lorenzelli et. al. developed an array of spatially distributed pH sensitive ISFETs to monitor the metabolic activity of a cell population [4]. To monitor electrical activity of neurons, an array ISFETs and gold microelectrodes was designed as shown in Figure 2.1. Metabolic and electrical activity was measured by the ISFET array while the gold microelectrodes were used to electrically stimulate neurons [15].

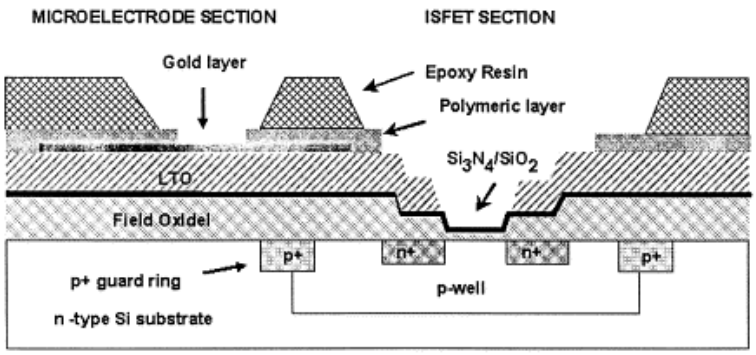


Figure 2.1: Cross-section of the ISFET sensor with the gold microelectrode layer [15].

2. Multi-parameter detection, such as flow velocity and diffusion coefficient of ions. Poghossian et. al. estimated flow velocity by measuring the time difference between the detection of H^+ ions by two spatially separated ISFETs [16]. The diffusion coefficient of ions was estimated by measuring the time taken for in-situ generated ions to be detected at each ISFET.

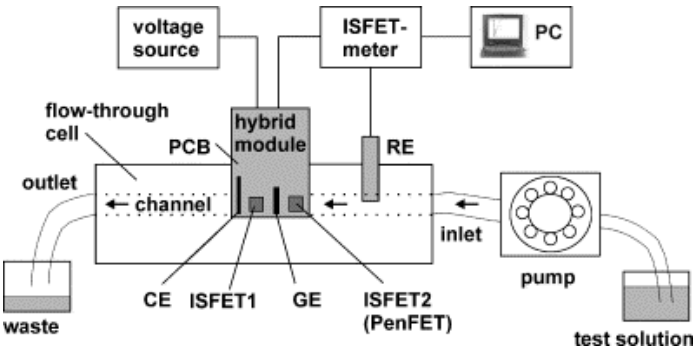


Figure 2.2: Schematic of the multi-parameter detection system using a pair ISFETs. A three electrode system maintains electrolyte potential and a voltage source is used for ion generation [16].

3. Monitoring enzyme-substrate and antigen-antibody reactions, as well as detecting DNA base pair hybridisation by modifying ISFETs into BioFETs [17]. BioFETs are fabricated from ISFETs by immobilising biological recognition molecules on the gate insulator. When a molecular binding reaction occurs, the change in gate potential results in a transduction mechanism.

   For example, Souteyrand et. al reported the fabrication of a DNA-modified FET or GenFET (Figure 2.3) by immobilising single-stranded DNA of known sequences onto the gate insulator of an ISFET [18]. DNA hybridisation changes the charge distribution in the channel and is detected as a change in threshold voltage.

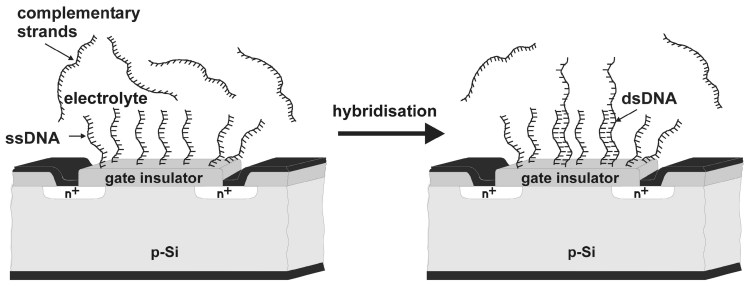


Figure 2.3: Structure of a GenFET used in DNA hybridisation detection [17].

2.2.2 Optical Sensors

Optical sensing can be performed by an active pixel sensor (APS) which contains an array of photodiodes and readout amplifiers. APS based imagers are cheap, low power and exhibit high fill-factor and quantum efficiency [19]. Unlike CCD sensors, APS allows optical sensing and image processing functions to be integrated on the same chip. Photodiode arrays are used in various applications, such as:

1. Detecting bioluminescence from bioluminometric single nucleotide polymorphins (SNP) typing assays. Kamahori et. al. reported the use of a photodiode array to detect bioluminescence from inorganic pryophosphates (PPi) released during primer extension [5] as illustrated in Figure 2.4. This process determines the bases present in SNPs.

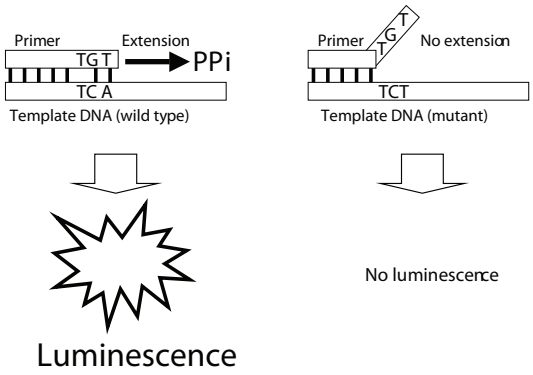


Figure 2.4: The principle of bioluminometric SNP typing assays [5]. A primer can extend its strand when its 3-terminus is matched to its target, but not when it is mismatched.

2. Detecting electrochemiluminescence (ECL) from electrically initiated chemiluminescent reactions between highly reactive molecules [20]. For example, Pal et. al. reported the use of a photodiode array to detect DNA hybridisation by labelling DNA strands with a ruthenium complex [21]. When DNA binding takes place, the ruthenium marker undergoes a series of reactions resulting light emission (Figure 2.5).

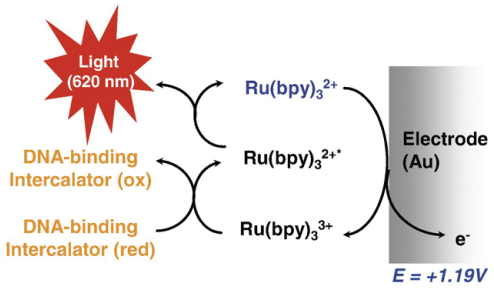


Figure 2.5: Electrochemical oxidation and reduction of a ruthenium complex $Ru(bpy)_3^{2+}$ used to tag DNA [22]. A series of redox reactions lead to photon emission which is detected by a photodiode array.

3. Detecting fluorescence during cell counting and sorting. Kruger et. al. developed a flow cytometry system which injects a single file cell flow through a microfluidic device where fluorescence activated cell sorting takes place [23]. Flow rate and cell sorting is controlled by a laser induced fluorescence detection system which uses an array of avalanche photodiodes (APD) as shown in Figure 2.6.

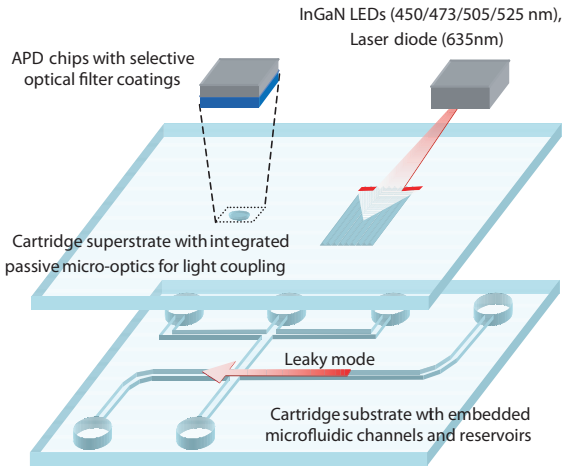


Figure 2.6: Microfluidic cartridge and laser induced fluorescence detection system [23].

2.2.3 Magnetic Actuators

Manipulation of biological molecules and cells can be performed on lab-on-chip devices through the use of paramagnetic micro-beads. Magnetic beads tagged with molecules and cells can be controlled by magnetic field patterns generated by an array of CMOS based inductive micro-coils [8]. Micro-beads offer a large surface to volume ratio for chemical binding, and are easily manipulated due to their intrinsic magnetic properties. Magnetic beads are used in a range of applications such as :

1. Cell labelling and magnetic separation where cells tagged with magnetic beads are separated using a magnetic separation device [24]. Morisada et. al. demonstrated immunomagnetic separation of cells by first coating the surface of micro-beads with a suitable antigen [25]. Cells with matching antibodies were then captured by the immunomagnetic beads and were separated by immobilising the beads magnetically as shown in Figure 2.7.

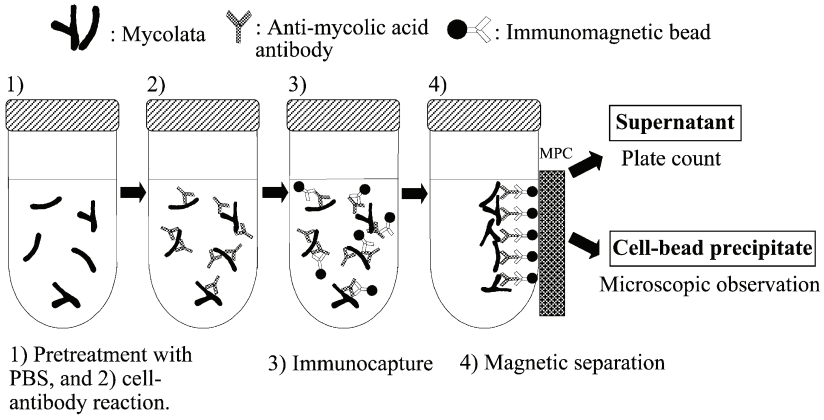


Figure 2.7: Immunomagnetic separation used to isolate mycolata (mycolic acid containing bacteria). Anti-mycolic acid bound to the mycolata are captured by immunomagnetic beads coated with a matching antigen. The cell-beads are separated from the supernatant magnetically [25].

2. Microparticle photometry where magnetic beads transported by an array of inductive micro-coils are detected by photodiodes (Figure 2.8). Lehmann et. al. demonstrated the use of an integrated inductive micro-coil and photodiode array to estimate particle velocity and magnetisation by observing particle departure and arrival between adjacent photodiodes in response to a changing magnetic field [26].

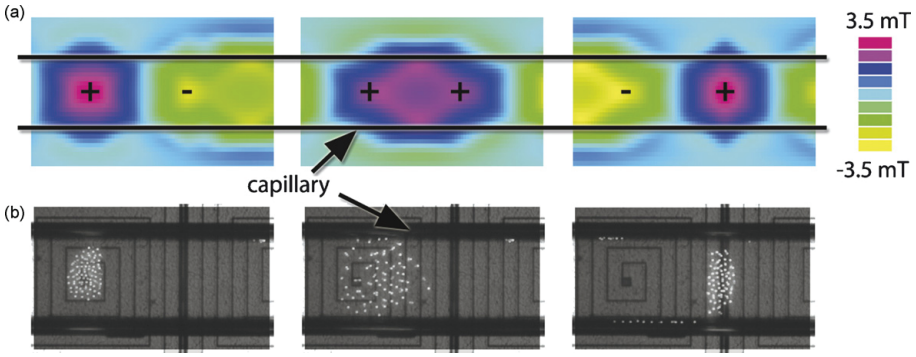


Figure 2.8: Magnetic bead transport across the CMOS chip. a) Simulation of the magnetic field: (+) is an attractive field while (−) is a repulsive field. b) Images of fluorescent magnetic particles overlaid with a photograph of the device [26].

3. Non-invasive manipulation and transportation of biological cells. Lee et. al. reported controlling the movement of cells tagged with magnetic beads by a user configurable magnetic field pattern generated from an array of inductive micro-coils. The CMOS chip allows high-speed generation of programmable magnetic fields, which offer high manipulability of magnetic beads.

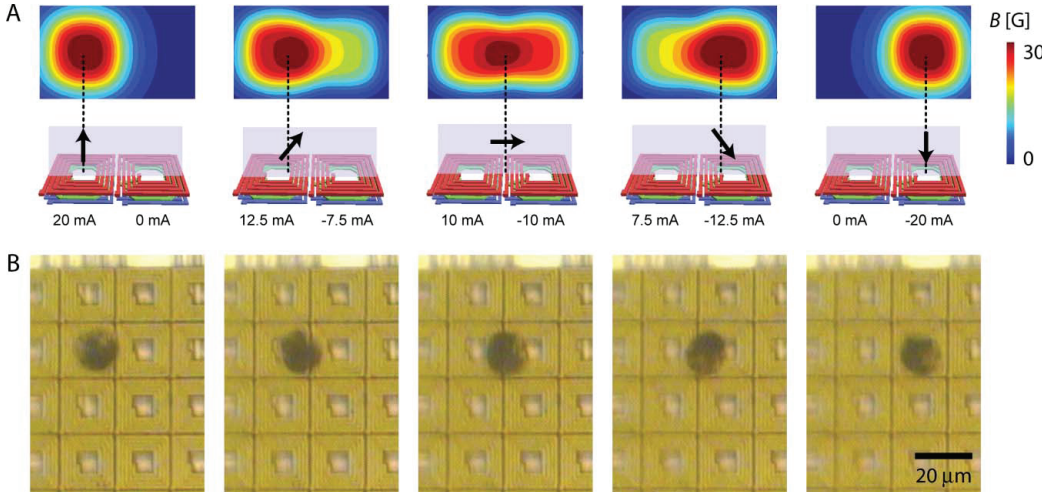


Figure 2.9: Programmable magnetic field patterns are generated to manipulate a cell across the inductive micro-coil array [8]. a) Simulated magnetic field with current distribution in micro-coils. The arrows indicate the magnetic field direction. b) Corresponding magnetic actuation of a single cell.

2.3 The T2P Project

This project involves developing a user platform for a novel CMOS-based lab-on-chip device (see Table 2.1 for comparison). The device incorporates chemical and optical sensors for the detection of pH and light, as well as inductive micro-coils to emit magnetic flux within a single pixel [7]. The system is scaled to form an 8×8 array (Figure 2.10) which allows for sensor calibration and a programmable magnetic field pattern for magnetic manipulation.

Year	Ref.	Tech. (μm)	Die Size (mm^2)	Array Size	Modalities	Application
2001	[15]	-	814	8×5	Chemical sensing	Monitoring cell population activity
2003	[13]	0.5	35.1	128×128	Neural sensing	Extracellular recording of neural activity
2004	[16]	-	6.4	2×1	Chemical sensing	Physical and chemical parameter measurement
2006	[14]	0.18	25	8×6	Optical sensing	Bioluminescence detection
2007	[8]	0.18	10	8×8	Magnetic acuation	Cell manipulation
2008	[26]	0.35	-	4×1	Optical sensing, magnetic actuation	Microparticle photometry
2009	[21]	-	400	12×12	Optical sensing	Electrochemiluminescence detection
2010	[7]	0.35	6.24	8×8	Opto-chemical sensing, magnetic actuation	-

Table 2.1: Comparative review of CMOS based lab-on-chip devices

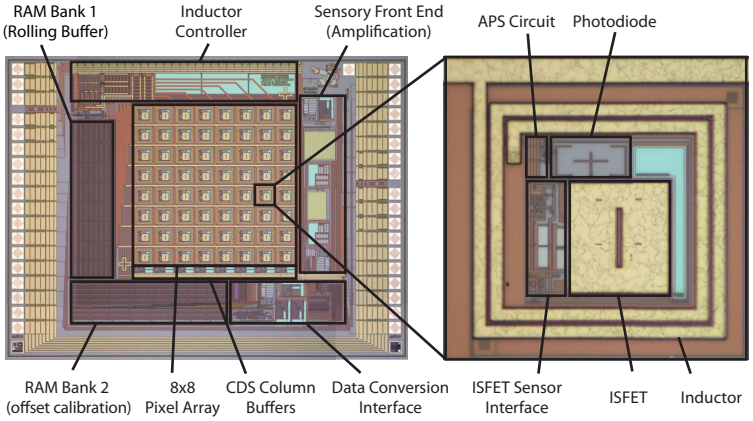


Figure 2.10: Micrograph of the CMOS-based lab-on-chip device and a multimodal pixel

The objective of the user platform is to integrate, control and process data from the CMOS-based lab-on-chip device for bio-analysis. The project specifications are listed below:

- Develop hardware for embedding the lab-on-chip device within a microfluidic platform
- Design instrumentation for communicating to and from the lab-on-chip device.
- Design a graphical user interface for real-time data acquisition of chemical and optical frames from the chip and the control of inductive micro-coils.
- Develop algorithms for calibration and mismatch reduction in the sensor array.
- Characterise the lab-on-chip sensor array sensitivity and actuation capabilities.

By combining magnetic manipulation with opto-chemical sensing, the lab-on-chip offers a platform for simultaneous monitoring of micro-scale bio-chemical assays with biomolecule and cell manipulation.

Chapter 3

System Architecture

The system architecture is shown in Figure 3.1. The lab-on-chip integrates three sub-systems for optical detection and chemical sensing, as well as generating micro-magnetic fields. The system uses a serial peripheral interface (SPI) protocol to load random access memory (RAM) data serially. This data contains $8 \times 8 \times 10$-bit words for chemical sensor calibration, 64×10-bit words to specify the magnetic field pattern and a 5-bit word which describes the programmable gains.

Reference circuits are also used to provide a 1μA PTAT current bias, 1.21V bandgap voltage reference and power on reset signals. The outputs are sampled using a 10-bit successive approximation analogue-to-digital converter (ADC) by interleaving the chemical and optical readout data. The chemical sensors are sampled during the second half clock period, while the optical sensors are sampled during the first half clock period. The schematic of the integrated pixel is shown in Figure 3.2.

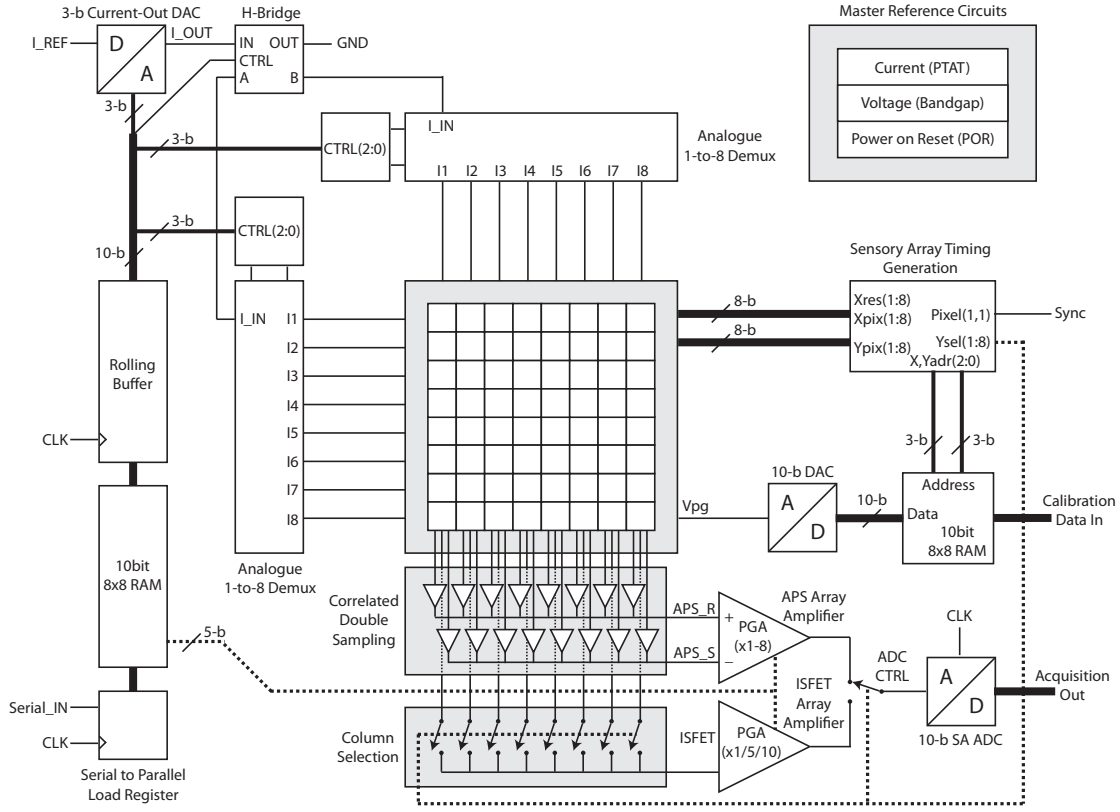


Figure 3.1: Top level architecture of the system [7].

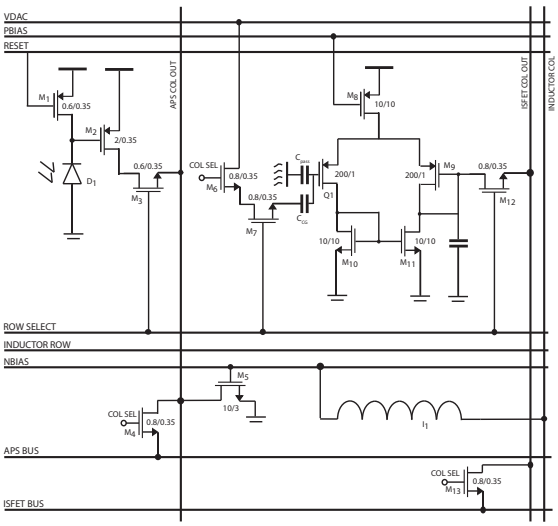


Figure 3.2: Schematic of the integrated pixel [7].

3.1 Inductive Micro-Coil Arrays

The lab-on-chip magnetic field pattern generator is programmed by writing binary data onto a 64×10-bit rolling buffer which cycles a 10-bit word onto the magnetic controller with the rising edge of each clock pulse. Instruction bits 9-7 and 6-4 represent the x and y coordinates respectively, bit 3 represents the polarity and bits 2-0 represent the current magnitude. A 3-bit digital-to-analogue (DAC) converter generates the current while its polarity is reversed using an H-bridge configuration. Bits representing pixel coordinates drive two de-multiplexers which bias the inductive micro-coil with a user defined current.

Manipulation and transportation of magnetic beads is achieved through a translational force generated from a magnetic field gradient. The magnetic force $\boldsymbol{F_m}$ acting on a superparamagnetic particle is given by Equation 3.1 [24] and its derivation can be found in Appendix A:

$$\boldsymbol{F_m} = \frac{V \triangle \chi}{\mu_0} (\boldsymbol{B} \cdot \boldsymbol{\nabla}) \boldsymbol{B} \tag{3.1}$$

where V is the magnetic particle volume, χ is the magnetic susceptibility, μ_0 is the magnetic permeability in vacuum and $\boldsymbol{B}$ is the magnetic flux density. The amount of magnetic flux required to produce this force is given by the Biot-Savart Law:

$$\boldsymbol{B} = \int \frac{\mu_0}{4\pi} \frac{I d\boldsymbol{I} \times \boldsymbol{r}}{r^3} \tag{3.2}$$

where I is the current through the micro-coil, $d\boldsymbol{I}$ is a vector whose magnitude is the length of the micro-coil and direction is the current's polarity. $\boldsymbol{r}$ is a unit vector pointing in the direction towards the point where the field is being computed and r is its corresponding scalor value [7]. By computing the forces required to move magnetic beads around an inductive micro-coil array, it is possible to achieve precise translocation of biological molecules and cells tagged with magnetic beads in various biological assays.

3.2 ISFET Arrays

The main operating princicple of the ISFET (Figure 3.3) is similar to that of the MOSFET, where the reference electrode acts as the gate and is inserted into aqueous medium. The gate oxide is replaced by an insulating membrane made of Al_2O_3, Ta_2O_5 or Si_3N_4, which traps H^+ ions at the oxide-solution interface. This alters the charge distribution in the channel, thus changing the threshold voltage of the ISFET as shown in Equation 3.3:

$$V_{th(ISFET)} = \gamma + 2.3 U_t \alpha \text{pH} + V_{th(MOSFET)} \tag{3.3}$$

where $V_{th(MOSFET)}$ is the threshold voltage of a MOSFET, γ is a grouping of pH independent potentials, U_t is the thermal voltage and $\alpha \in [0, 1]$ is a scaling factor describing the reduction in pH sensitivity [27].

By monitoring the threshold voltage, we are able to estimate the solution's pH.

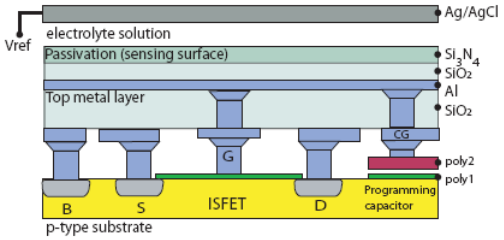


Figure 3.3: Layout of the PG-ISFET [28]

Despite the many advantages of using ISFETs, they are prone to many sensor non-idealities such as threshold voltage mis-match and drift [29]. These non-idealities are due to trapped charge which reside in the passivation layer or floating gate [30]. In order to reduce device mis-match, programmable gate (PG) ISFETs have been developed to enable calibration of the device operating point [28].

Calibration is achieved by applying a voltage bias to a control gate capacitor which is coupled to the floating gate of the PG-ISFET (Figure 3.3). During calibration, the programmable gate inputs are loaded in the form of a 64×10-bit binary lookup table. When the sensor array is polled, each pixel is biased accordingly from a DAC which reads in the corresponding 10-bit instruction from the lookup table. This allows the system to eliminate sensor non-idealities as discussed in Appendix B.

3.3 Photodiode Arrays

A standard APS pixel architecture (Figure 3.4) consists of a photodiode, reset transistor M_1, a source-follower readout transistor M_2 and a select transistor M_3. During photo-detection, a reset signal is asserted onto M_1 to initialise and reverse-bias the photodiode. When the photodiode is illuminated, it loses charge stored in the depletion region at a rate proportionate to the illumination intensity. By integrating the photodiode voltage over a fixed sample time, we are able to obtain a signal which is proportionate to the incident illumination level [31].

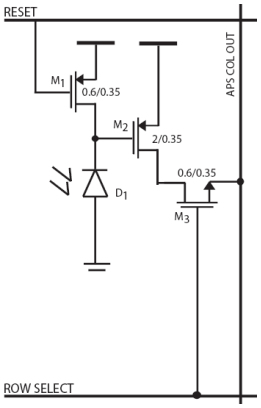


Figure 3.4: Schematic of an APS pixel [7]

One limitation of using APS sensors is the presence of reset or $^{kT}/_{C}$ noise [32]. To improve the signal-to-noise ratio, correlated double sampling (CDS) can be used to suppress $^{kT}/_{C}$ noise, $^{1}/_{f}$ noise and threshold voltage variations from the source-follower readout transistor. CDS works by taking two samples during pixel readout. The first sample taken when the pixel is in the reset state contains noise only. The second sample which is taken when the photodiode voltage has been discharged contains both signal and noise. The samples are subtracted, eliminating $^{kT}/_{C}$ noise and attenuating $^{1}/_{f}$ at low frequencies [33]. This improves the signal-to-noise ratio of APS sensor readouts, resulting in better image quality.

Chapter 4

Hardware Implementation

4.1 Lab-on-Chip Development Board

A printed circuit board (Figure 4.1) integrating various hardware components and the lab-on-chip device was designed using OrCAD. The development board allows the user to control data to and from the chip via a programmable PIC micro-controller from a MATLAB graphical user interface (GUI).

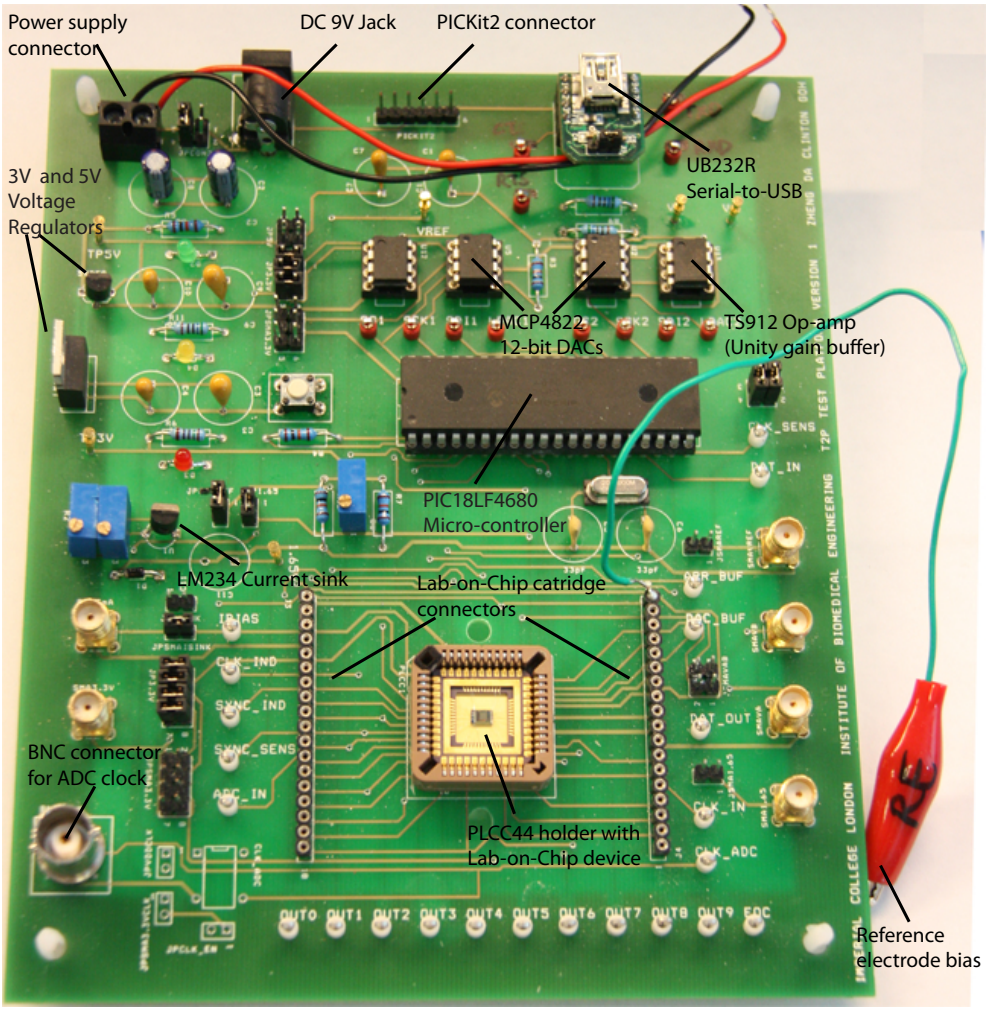


Figure 4.1: PCB incorporating microcontroller, lab-on-chip and other peripheral devices

4.1.1 PIC Microcontroller

A Microchip 18LF4680 PIC microcontroller was used for integrating and controlling data between the lab-on-chip device and MATLAB. A microcontroller programmer (Microchip PICKit2) was included for easy programming, while USB-to-Serial communication between the microcontroller and MATLAB was established using a UB232R module. In addition to setting up hardware receive and transmit pins, software handshake between the microcontroller and MATLAB was enabled by interfacing the request-to-send (RTS) and clear-to-send (CTS) pins.

Software SPI pins were allocated to program the lab-on-chip RAM and voltage references. To enable real time data acquisition, the end-of-conversion (EOC) signal from the lab-on-chip was interfaced to a hardware external interrupt pin on the PIC to allow sampling of digital outputs once the ADC has completed conversion.

A state machine is programmed within the PIC micro-controller to perform the following functions:

- Program external digital-to-analogue converters (DAC) to provide reference voltages.
- Program the lab-on-chip magnetic field pattern, PG-ISFET gate bias and programmable gain amplifiers (PGAs).
- Generate clock signals for generating the magnetic field pattern.
- Generate clock signals to poll through the sensory pixel array.
- Sample the digital output word from the lab-on-chip during real time data acquisition.
- Receive state machine commands and transmit stored data to MATLAB.

4.1.2 Voltage References

Two DACs (Microchip MCP4821 and MCP4822) were used to provide reference voltages for the lab-on-chip ADC and an external reference electrode. The DACs are programmed via a SPI protocol, and offer up to 12-bit resolution. The output voltage generated by each 12-bit word is given by Equation 4.1.

$$V_{OUT} = \frac{2.048V \times G \times D_N}{2^n} \tag{4.1}$$

where G is the programmable gain ($1\times$ or $2\times$), D_N is the digital input value and n is the number of bits of resolution ($n = 12$). To improve the stability of reference voltages generated by the DACs, the outputs of both DACs are fed through rail-to-rail unity gain buffers.

4.1.3 Inductive Micro-coil Current Sink

An adjustable three terminal current source (National Semiconductor LM234) was used to provide a current reference by drawing 3mA from the lab-on-chip. The magnitude and polarity of the current through each inductive micro-coil is determined by a 3-bit DAC based on a binary weighted current mirror [34] and an H-bridge configuration respectively [7]. The schematic of the current sink is shown in Figure 4.2 and the amount of current drawn can be computed using Equation 4.2.

$$I_{SINK} = \frac{0.134V}{R_2} \quad , \quad \frac{R_1}{R_2} = 10 \tag{4.2}$$

4.1.4 ADC Clock

An external 1MHz oscillator was used to provide a clock signal for the lab-on-chip's internal ADC. The clock signal can also be provided by a signal generator via a Bayonet Neill-Concelman (BNC) connector.

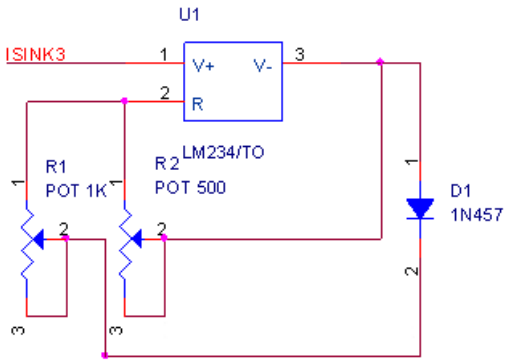


Figure 4.2: Adjustable three terminal current source

4.2 Lab-on-Chip Cartridge

A secondary printed circuit board was designed for wire bonding and encapsulating the lab-on-chip device and to provide an elevated platform for experiments to be conducted on. Wire bonding was performed between the lab-on-chip and two rows of bond pads on each side of the cartridge. The lab-on-chip was then encapsulated using glob-top resin to provide mechanical support for bond wires and to prevent contamination of the device from chemicals. For micro-fluidic integration, the lab-on-chip was encapsulated with parylene to provide a chemically resistant coating with a completely homogenous surface.

The lab-on-chip cartridge shown in Fig 4.3 is interfaced to the main instrumentation board via corresponding connectors and can be used in liquid based assays or with a micro-fluidic platform. A fluidic well was used to contain fluid for biochemical based assays. The well was constructed by adhereing the exposed tip of a falcon tube to the glob-top resin surrounding the encapsulated chip. The well contains up to 2ml of fluid and allows for simultaneous USB microscope imaging.

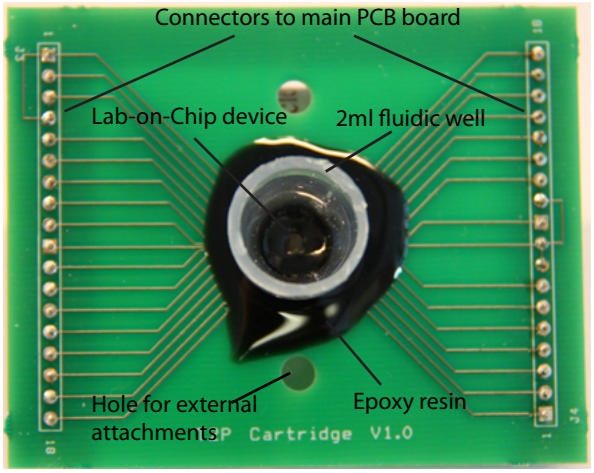


Figure 4.3: Lab-on-chip Cartridge

Chapter 5

Software Implementation

A series of MATLAB based GUIs were developed to program the lab-on-chip, perform real time data acquisition, calibrate the sensor array and view stored image frames on the PC. Data communication between MATLAB and the lab-on-chip was established through the microcontroller, based on a standard UART interface. A finite state machine protocol facilitated the transfer of instructions and information between MATLAB and the microcontroller.

5.1 Graphical User Interface Main Menu

All GUIs are accessed through a main menu shown in Appendix D. The GUI allows the user to set up serial communication between the microcontroller and PC, as well as to configure the image acquisition device via a preview window.

From the main menu, the user can call separate sub-GUIs to program the magnetic field pattern, initialise, calibrate or perform real time data acquisition on the lab-on-chip, and view previously stored data. Hardware device parameters specified within the main menu are passed to these GUIs so as to allow the user to communicate with the lab-on-chip via the secondary GUIs.

5.2 Magnetic Field Pattern Programmer

The magnetic field pattern is generated from a 64×10-bit RAM rolling buffer where each 10-bit word represents the x-coordinate (bits 9-7), y-coordinate (bits 6-4), polarity (bit 3) and magnitude (bits 2-0). The GUI shown in Appendix F allows the user to program each RAM element which is displayed in a user interactive table.

User specified data is encoded into digital bits and the amount of current drawn through each inductive micro-coil is displayed in a 3D plot. The GUI also allows the user to simulate the current drawn through each micro-coil during RAM playback. Once the magnetic field pattern has been programmed, data bits are pre-processed and stored in a 128×8-bit array to program the lab-on-chip.

Data pre-processing is required due to the following reasons:

1. The order in which binary bits are clocked in follows a meander pattern as shown in Figure 5.1. Data bits are arranged and aligned by reversing bits which belong to even rows when transforming a MATLAB 8×8 matrix into a 1×64 vector to be sent to the microcontroller.

2. The microcontroller only stores data as an 8-bit or 16-bit binary word. Each 10-bit instruction is thus separated into its 2 most significant bits and 8 least significant bits and stored as a pair of 8-bit binary words.

3. Since the command `fgets` recognises `char` 13 and 10 as terminators, it is possible that the data vector sent from MATLAB is prematurely terminated. To deal with this, extra bits in the first 8-bit word of each instruction are used to identify these special characters.

Using the above data processing protocol, user defined data is first sent from MATLAB to the micro-controller as a series of bits without any information loss. Binary bits are then decoded and clocked into the lab-on-chip such that each 10-bit instruction is aligned correctly within the internal RAM buffer.

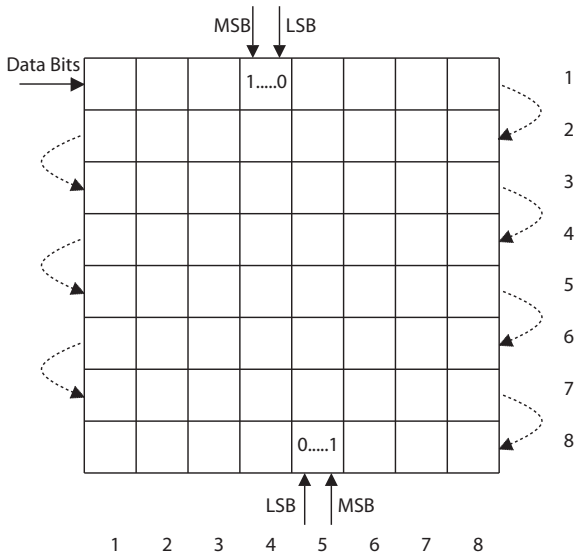


Figure 5.1: Bits clocked into the lab-on-chip RAM

5.3 Lab-on-Chip Initialisation

Initialisation of the lab-on-chip involves setting up experimental parameters such as ADC reference voltages, ISFET reference electrode bias and PGA gains. The layout of this GUI module is shown in Appendix E. The user specifies experimental parameters which are encoded and sent as state machine instructions to the micro-controller.

Encoding reference voltages is based on Equation 4.1, where each user defined voltage is used to compute the DAC gain G and its corresponding 12-bit word D_N. G is set to 1 for $V_{out} < 2.048$V and to 2 otherwise. Following the SPI protocol for MCP4822 DACs, a 16-bit instruction containing encoded bits for G and D_N is generated and sent to the micro-controller with its corresponding state machine command.

Encoding optical gain is based on a 3-bit binary word ranging from 1 to 8. Chemical gain takes on discrete values of 1, 5, 6 and 10, which are encoded by 2-bit words `00`, `01`, `10` and `11` respectively. The 5-bit instruction is formed by appending bits representing chemical gain to that of optical gain and is sent with a state machine command to the micro-controller.

Lab-on-chip pointers can also be initialised via state machine commands sent from the GUI. During power up, the position of RAM and sensor array pointers are unknown. Utilising lab-on-chip synchronising signals, the microcontroller generates interrupt driven clock signals to initialise lab-on-chip pointers at known locations in the RAM buffer and pixel array. This allows for easy data tracking during ISFET calibration and real time data acquisition.

5.4 Real Time Data Acquisition

Acquiring chemical and optical data in real time from the lab-on-chip device involves polling the sensor array and displaying acquired data in MATLAB. This can be performed independently or in parallel with the generation of a spatiotemporal magnetic pattern. An external image acquisition device can also be used to monitor the movement of magnetic beads or the state of a micro-fluidic system. The GUI shown in Appendix E implements real time data acquisition and allows the lab-on-chip to be calibrated and initialised.

Real time data acquisition combining both sensor array polling and magnetic field pattern generation was implemented using the following software control protocol:

1. To start, MATLAB sets the RTS pin to high and sends a state machine command to the microcontroller. Image acquisition is enabled and captured frames are displayed in MATLAB.
2. If the RTS pin is low, the microcontroller disables all interrupts and stops both the data and image acquisition process.
3. Else, the microcontroller generates an interrupt driven clock signal to cycle through the RAM rolling buffer of the magnetic pattern generator.
4. MATLAB sends a state machine command to the microcontroller to acquire data from a single frame and waits for the data to be ready.
5. The microcontroller generates an interrupt driven clock signal to poll the sensor array.
6. Each end-of-conversion (EOC) pulse drives an external interrupt on the microcontroller. The 10-bit digital output word from the lab-on-chip is sampled and stored in memory.
7. After all 64 pixels are polled, the microcontroller pushes data bits onto the PC serial buffer and pulses the CTS pin, indicating that data is ready to be acquired.
8. MATLAB acquires, processes serial data, and displays chemical and optical information as images.
9. To stop, MATLAB sets the RTS pin to low. Else, Go to Step 2.

Sensor array polling can also be done independently of magnetic field pattern generation by simplifying the above protocol to include steps 4 to 8 in a loop. Likewise, the magnetic field pattern can be generated independently by only considering steps 1 to 3 and 9. This gives the user full control over acquiring data with or without simultaneously generating the programmed magnetic field pattern.

The GUI also allows the user to selectively bias the inductor current of individual pixels while cycling through the RAM rolling buffer. The user first specifies the current magnitude and polarity of a pixel to be biased. The RTS pin is then asserted to a low state and the magnetic field pattern is programmed into the lab-on-chip. Data acquisition resumes automatically and the software protocol initialises from step 1. By biasing individual pixels, magnetic bead movement can be controlled by the user in real time.

The magnetic field pattern playback frequency can also be varied via the GUI. The user specified clock frequency is first encoded within the state machine instruction bits and streamed to the microcontroller. The instruction bits are decoded and then used to alter the interrupt routine which generates the clock signal.

5.5 Image and Data Viewer

Acquired data can be saved from the real time data acquisition GUI as a `*.mat` file and retreived via the image and data viewer GUI shown in Appendix G. This GUI provides the user with a fast and efficient way of visualising images through the following modes:

1. 2D Image: Chemical and optical information is displayed as an 8×8 pixel, 2 dimensional image. This mode is used to visualise chemical and optical intensities with respect to the chip surface.
2. 3D Image: Chemical and optical data is displayed as an $8 \times 8 \times V_{out}$, 3 dimensional image. This mode allows the user to visualise chemical and optical intensity contours.
3. Montage: Chemical and optical images are displayed in the form of a montage. This mode is used to compare 2D images from consecutive frames.
4. Line: Chemical and optical information is displayed as a time series. This mode allows the user to monitor time varying changes in chemical and optical intensities. The data can be filtered to remove noise and time varying readouts from multiple pixels can be displayed simultaneously.

5.6 Chemical Sensor Calibration

One advantage of using PG-ISFET arrays is the capability to calibrate out sensor non-idealities. The calibration process aims to bias the ISFET array at mid supply range (1.65V) when the device is exposed to an aqueous solution of pH = 7. Like most optimisation problems, calibration is achieved when the mean squared error between the ISFET output and the operating point is minimised:

$$f(x) = \frac{1}{N}\sum_{i=1}^{64} \left(V_{out}(x_i) - V_{op}\right)^2 \tag{5.1}$$

where the mean squared error $f(x)$ is a function of the control gate voltage $x \in \mathbb{R}^{64}$, $V_{out}(x_i)$ is the output of the each pixel in the ISFET array and V_{op} is the operating point. f is minimised when:

$$\nabla f(x) = \frac{2}{N}\sum_{i=1}^{64} \left(V_{out}(x_i) - V_{op}\right) \nabla V_{out}(x) = 0 \tag{5.2}$$

A general algorithm which can be used to optimise the problem involves searching for a sequence $\{x_k\}$ which starts from initial point x_0, while converging to the set $\Omega = \left\{x \in \mathbb{R}^{64} : \nabla f(x) = 0\right\}$ as described below:

1. Fix a point $x_0 \in \mathbb{R}^{64}$ and set $k = 0$.
2. If $x_k \in \Omega$, STOP. Else, compute a direction of research $d_k \in \mathbb{R}^{64}$.
3. Compute a step $\alpha_k \in \mathbb{R}$ along d_k.
4. Let $x_{k+1} = x_k + \alpha_k d_k$. Set $k = k + 1$ and go to step 2.

The gradient algorithm was implemented for ISFET calibration, where the anti-gradient: $d_k = -\nabla f(x)$ is selected as the research direction, and α_k is computed by any line search method such that $f(x_k + \alpha_k d_k) \leq f(x_k)$. Hence, d_k is the offset of the ISFET output multiplied by $\nabla V_{out}(x)$. Since $V_{out}(x)$ is the floating gate voltage of the ISFET as discussed in Appendix B, then $\nabla V_{out}(x)$ is the ratio of ISFET capacitances multiplied by the chemical gain, and is a non-zero constant. Hence, f is minimised only when $V_{out}(x_i) = V_{op}$. The algorithm is simplified by grouping α_k and $\nabla V_{out}(x)$ together as $\rho_k < 1$, eliminating the need for a line search algorithm.

Although calibration aims to satisfy Equation 5.2, this may be impossible due to noise and quantisation error from the lab-on-chip's 10-bit DAC which offers a finite resolution of 32mV on a gain of 10. Searching for a converging sequence may result in oscillations about the operating point.

To ensure convergence, the optimisation problem is relaxed to allow $\{x_k\}$ to converge to the set $\Phi = \left\{x \in \mathbb{R}^{64} : f(x) < \epsilon\right\}$. The algorithm terminates once the mean squared error is below a user defined threshold value ϵ as described below:

1. Program the lab-on-chip with an initial control gate voltage x_0 =1.65V and set $k = 0$.
2. Poll the sensor array and obtain data from a single frame. Compute $f(x)$. If $f(x) < \epsilon$, STOP.
3. Else, set $d_k = -\left(V_{out}(x) - V_{op}\right)$.
4. Compute new control voltages $x_{k+1} = x_k + d_k \rho_k$ and set $k = k + 1$.
5. Re-program the lab-on-chip with new control gate voltages and go to Step 2.

The calibration algorithm is now dependent on two user defined parameters: ρ_k and ϵ. To increase the speed of convergence, we can choose a larger ρ_k and ϵ. Choosing a small ρ_k improves algorithm accuracy but decreases the speed of convergence. Likewise, choosing a large ϵ leads to a faster speed of convergence at the expense of accuracy. Hence, choosing an appropriate step size ρ_k and error threshold ϵ can optimise the speed of convergence and accuracy of the calibration algorithm and is discussed in Section 6.5.

Chapter 6

System Verification

6.1 Pointer Initialisation

Lab-on-chip pointers were initialised by utilising the SYNC_IND and SYNC_SENS signals which are asserted high when pointing at the first ram element and pixel $(1, 1)$ respectively. The lab-on-chip was initialised by generating a clock signal until both pins change from a logic state of 0 to 1 (Figure 6.1 and 6.2). Since it takes 4 clock cycles to poll from one pixel to the next, three extra clock pulses were generated after SYNC_SENS turned high. This sets the sensor pointer at pixel $(1, 1)$ and prevents the optical saturation of pixel $(1, 2)$. Once initialisation is complete. CTS is asserted high and the microcontroller waits for a corresponding handshake from the PC.

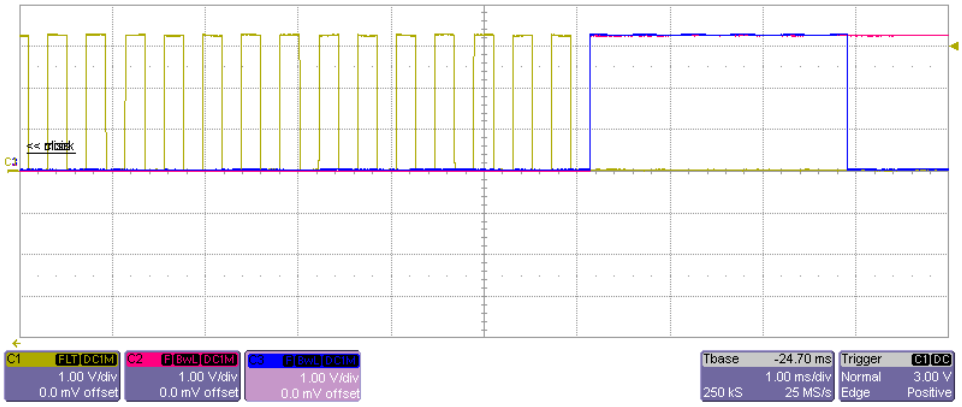


Figure 6.1: Initialising the lab-on-chip RAM pointer. The clock signal is the yellow trace, SYNC_IND is the red trace and CTS is the blue trace.

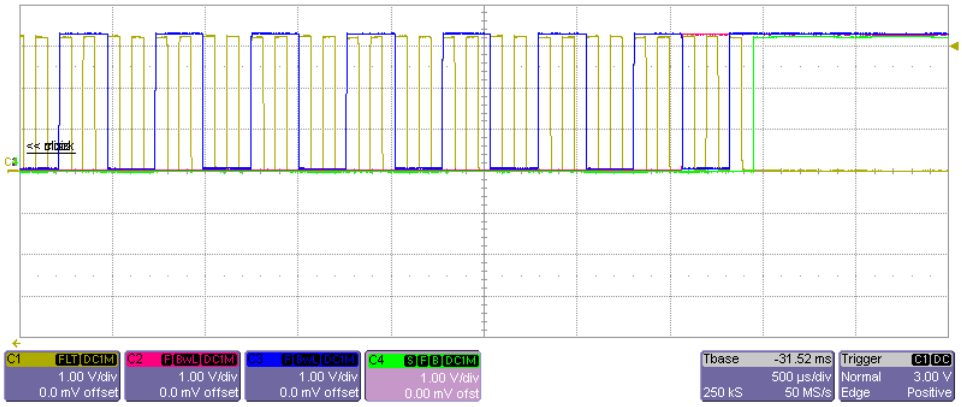


Figure 6.2: Initialising the lab-on-chip pixel pointer. The clock signal is the yellow trace, ADC_IN is the blue trace, SYNC_SENS is the red trace and CTS is the green trace.

6.2 ADC Reference Voltage Initialisation

The microcontroller was used to program the 12-Bit DACs via an SPI interface, after receiving a state machine command and encoded 16-bit instruction from MATLAB. Figure 6.3 shows the reference voltage programmed as a discrete triangle function over time. Since the selected gain G must be set to $\times 2$ for $V_{OUT} > 2.048\text{V}$, the LSB size increases from $^{2.048\text{V}}/_{4096}$ for $V_{OUT} < 2.048\text{V}$ to $^{4.096\text{V}}/_{4096}$ for $V_{OUT} \geq 2.048\text{V}$.

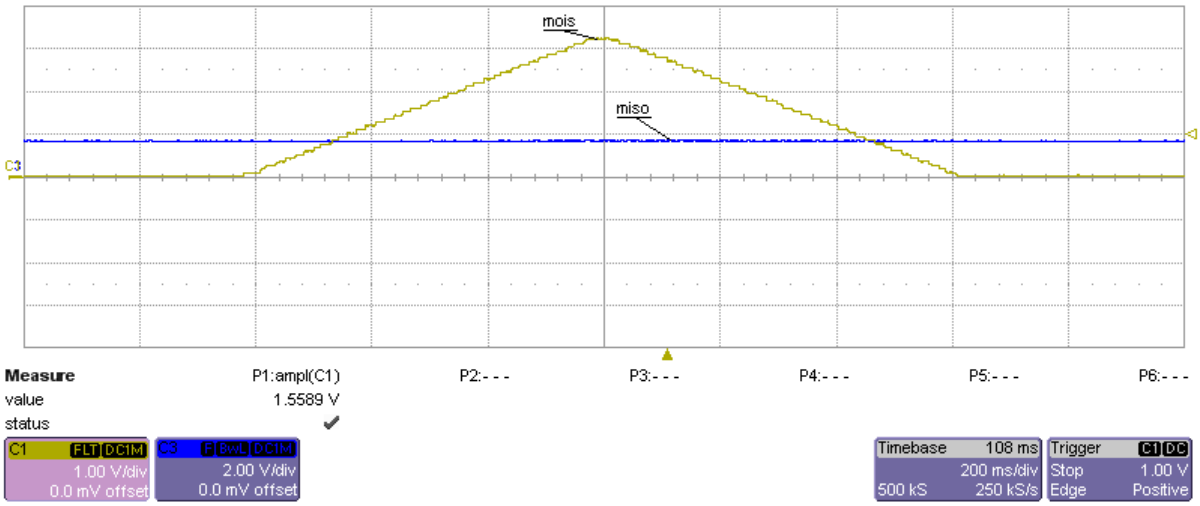


Figure 6.3: Reference voltage programmed as a triangle function

6.3 Magnetic Field Pattern Programming

6.3.1 3-Terminal Current Sink

The current sink was tested with a Keithley source-meter. The isolated current sink was found to be capable of sourcing 3mA from the power supply rails, but failed to do so from the IBIAS_IND pin of the lab-on-chip. To investigate the cause of this, the source-meter was used to pull varying currents and the corresponding voltage was measured. Figure 6.4 shows the characteristic I-V curve obtained.

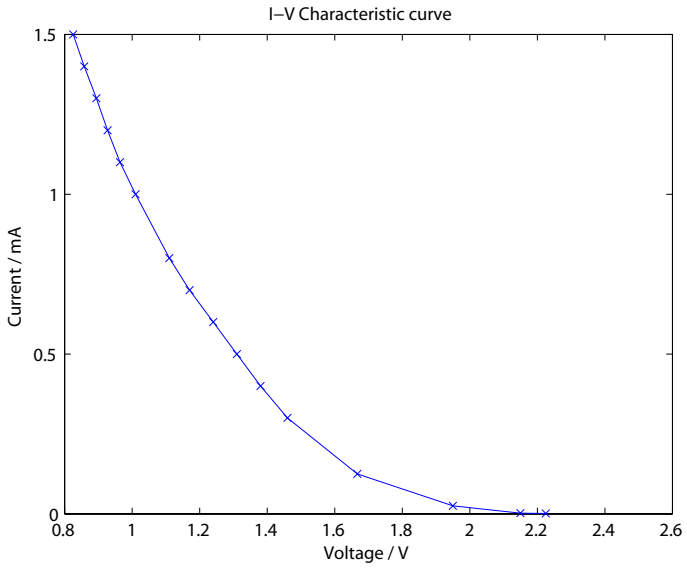


Figure 6.4: I-V characteristic curve of the IBIAS_IND pin

It can be observed that as the curent increases, the voltage decreases in a quadratic manner. This is due to internal lab-on-chip circuitary where the current is drawn from two gate-source diode connected PMOS transistors. Hence there is a $2V_{GS}$ potential drop at where the current sink is connected to: $V_{IND} = V_{DD} - 2V_{GS}$. As the current drawn through each PMOS device increases, the gate-source voltage increases according to:

$$I_{DS} = \frac{\mu_p C_{ox}}{2} \frac{W}{L} \left(V_{GS} - \|V_{th,p}\|\right)^2 \tag{6.1}$$

where I_{DS} is the current through each PMOS device, μ_p is the mobility of holes, C_{ox} is the oxide capacitance, W/L is the ratio of gate width to length, V_{GS} is the gate source voltage and $V_{th,p}$ is threshold voltage.

Increasing I_{DS} results in decreasing V_{IND} values, as observed in Figure 6.4. The operating voltage of the current sink is limited by the $2V_{GS}$ term, which then limits the maximum current drawn. Hence it is only possible to draw up to 1.5mA with an operating voltage of approximately 0.8V.

In order to allow for currents greater than 1.5mA to be drawn from the lab-on-chip device, an external integrated circuit current mirror can be used with the implemented LM234 current sink. This allows the current sink to be biased from the supply rails, avoiding the problem of limited voltage swing.

6.3.2 Magnetic Field Pattern Playback

A user defined magnetic field pattern was programmed using MATLAB and loaded onto the lab-on-chip via the microcontroller. The amount of current drawn through each pixel was made to follow a triangle wave function. A state machine command was then sent to the microcontroller, which generated an interrupt driven clock signal to cycle through each instruction in the RAM rolling buffer. The total amount of current drawn from power supplies was monitored and is shown to follow a similar triangle wave pattern in Figure 6.5.

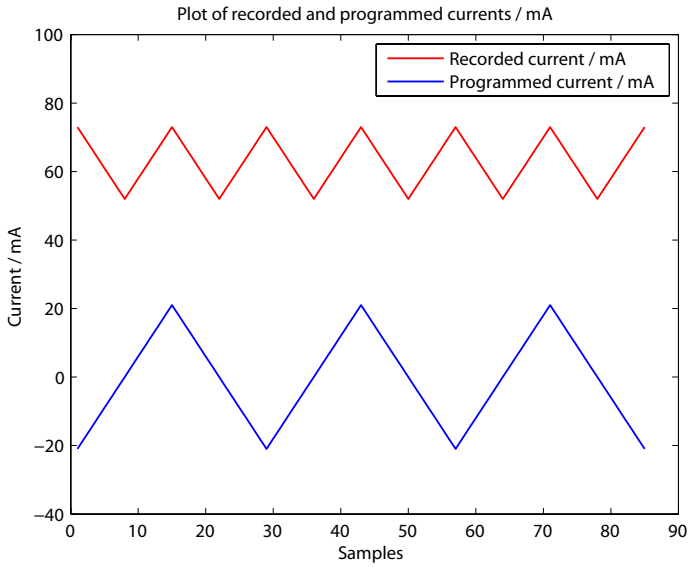


Figure 6.5: Total current drawn from the power supplies. A DC offset is present due to current drawn from other devices present on the development board.

It can be observed that the recorded current I_{OUT} is a function of programmed current I_{IN}:

$$I_{OUT} = |I_{IN}| + I_{DC} \tag{6.2}$$

The $|I_{IN}|$ term originates from current steering circuits which reverse the polarity of the current through each pixel, while I_{DC} is due to power consumption of other peripheral devices on the development board.

6.4 Real Time Data Acquisition

Data acquisition from the sensor array was implemented by polling all 64 pixels and sampling output bits on the lab-on-chip via an EOC signal driven interrupt routine. Once the microcontroller has stored data from a single frame, it pushes the data onto the PC serial buffer. After MATLAB processes and displays acquired data as an image, the process is repeated again. Figure 6.6 illustrates the data acquisition process.

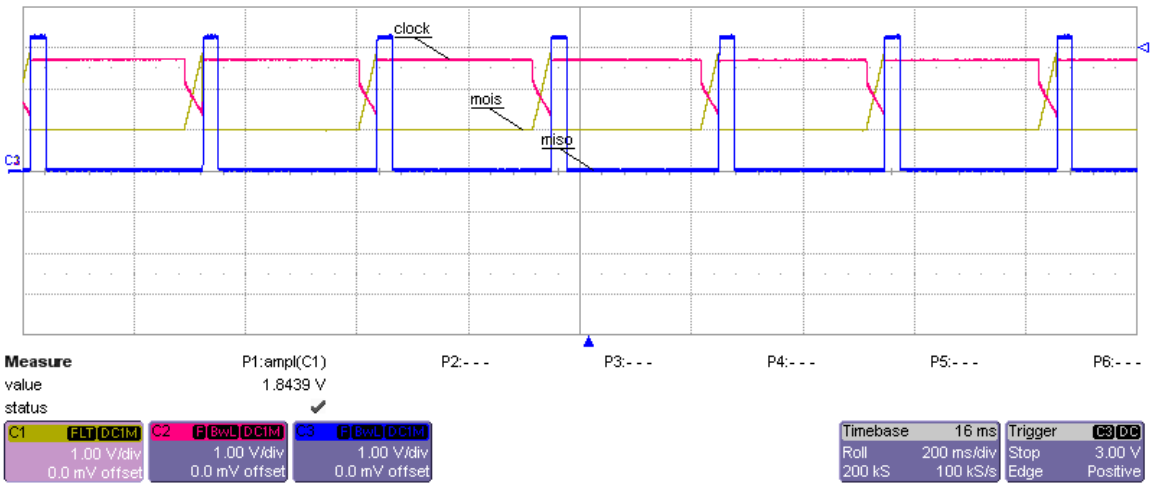


Figure 6.6: Real time data acquisition from the lab-on-chip sensors. The sensor array output is the red trace, the control gate voltage is the yellow trace and the CTS signal is the blue trace.

It can be observed that the bottle-neck of the data acquisition process is the time taken to process and display acquired data in MATLAB. This limits the frame rate to approximately 6Hz, which is slow in terms of modern day imaging systems. This can be improved by using a PC with a faster processor, or by migrating the GUI from MATLAB into C.

6.4.1 Chemical Data Acquisition

The functionality of the PG-ISFET array was verified by programming a user-defined, monotonically increasing and decreasing control gate vector into the lab-on-chip. When each pixel is addressed, a 10-bit word is decoded and used to bias each PG-ISFET. Figure 6.7 shows the sensor readout from the lab-on-chip, while Figure 6.8 shows the acquired chemical images for different control gate bias vectors.

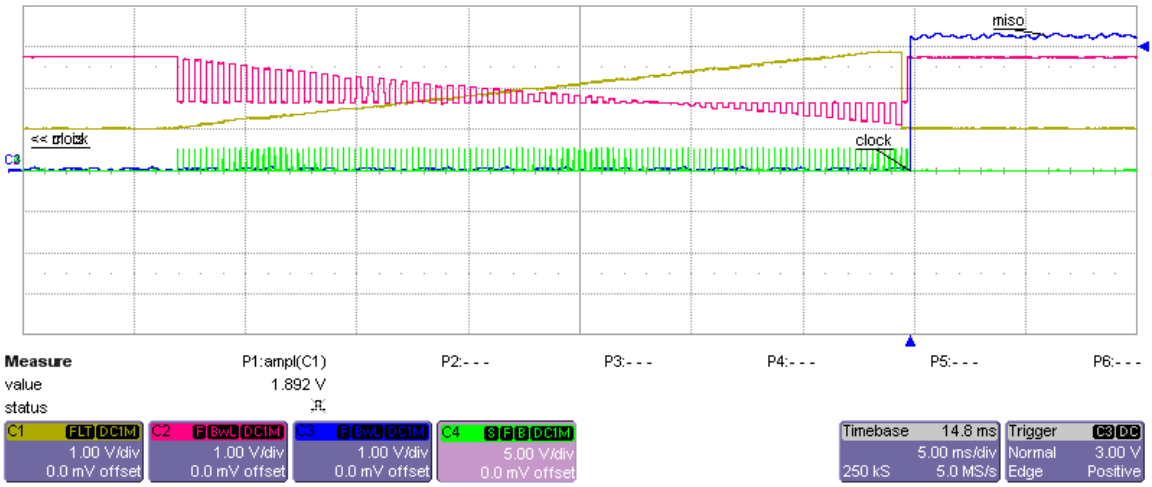


Figure 6.7: Polling a frame from the sensor array. The output of each pixel is the red trace, the control gate voltage is the yellow trace, the CTS signal is the blue trace and the EOC signal is the green trace.

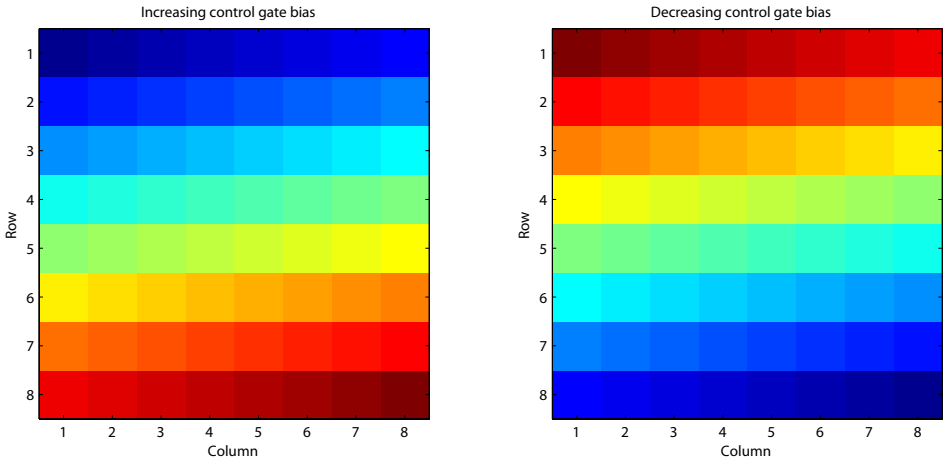


Figure 6.8: Chemical images acquired from monotonically increasing and decreasing control gate biases.

Since the chemical output is connected to the inverting input of an op-amp and compared against the mid-supply potential of 1.65V, the chemical readout decreases for increasing values of control gate voltage. However, both lines do not cross at 1.65V. Despite the presence of a DC offset, Figure 6.7 shows that the operating point of each PG-ISFET can be easily defined.

The chemical readout was found to saturate quickly when maximum gain is used. However, we expect the chemical readout to have a range of ± 0.4V, since the maximum Nernstian sensitivity of an ISFET is 58.2mV/pH [35]. To improve digital resolution, the PGAs would be set to maximum gain and the ADC reference voltage range reduced.

6.4.2 Optical Data Acquisition

The functionality of the photodiode array was verified by focusing a spot of light using a Pentax H1212B lens. Light intensity was varied and the light stimulus was positioned across the device. Optical data was extracted from each frame and displayed as a sequence of images. The photodiode array responded proportionately to varying illumination intensities as shown in Figure 6.9 and was also able to detect a translation in optical stimulus across the chip surface as shown in Figure 6.10.

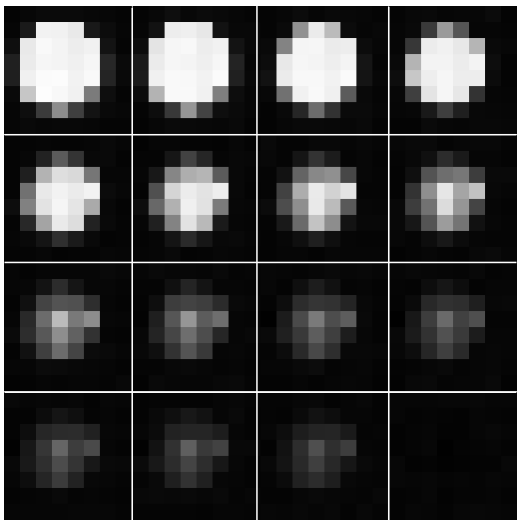
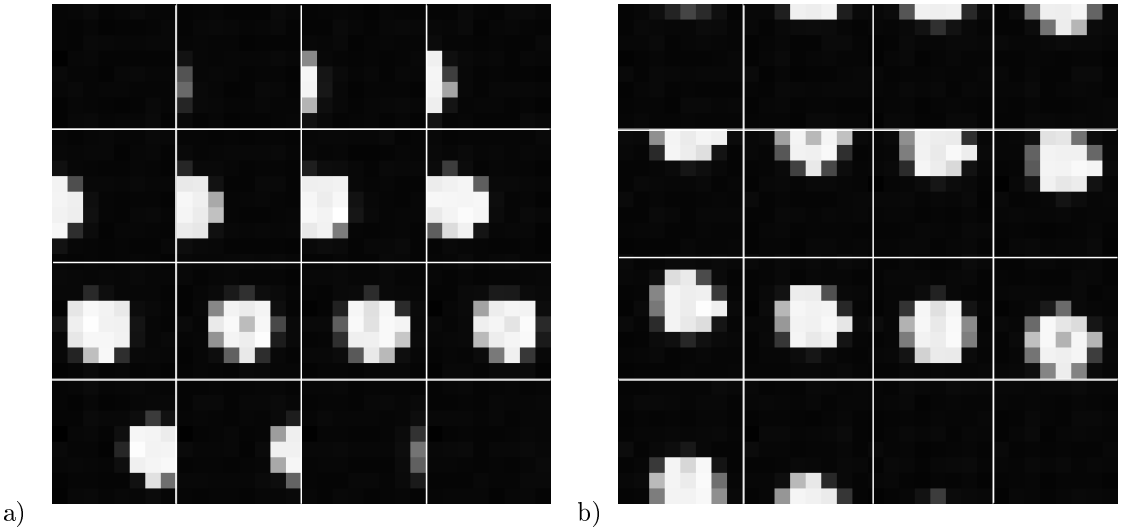


Figure 6.9: Montage of optical images showing a spot of light with varying illumination intensities



a) b)

Figure 6.10: Optical images of a spot of light moving across the lab-on-chip. a) The spot moves from left to right. b) The spot moves from top to bottom.

6.4.3 Real Time Data Acquisition with Magnetic Field Pattern Playback

Simultaneous sensor array data acquisition with magnetic field pattern playback was carried out on the lab-on-chip device. Figure 6.11 shows the data acquisition process combined with a clock signal to cycle through the RAM rolling buffer.

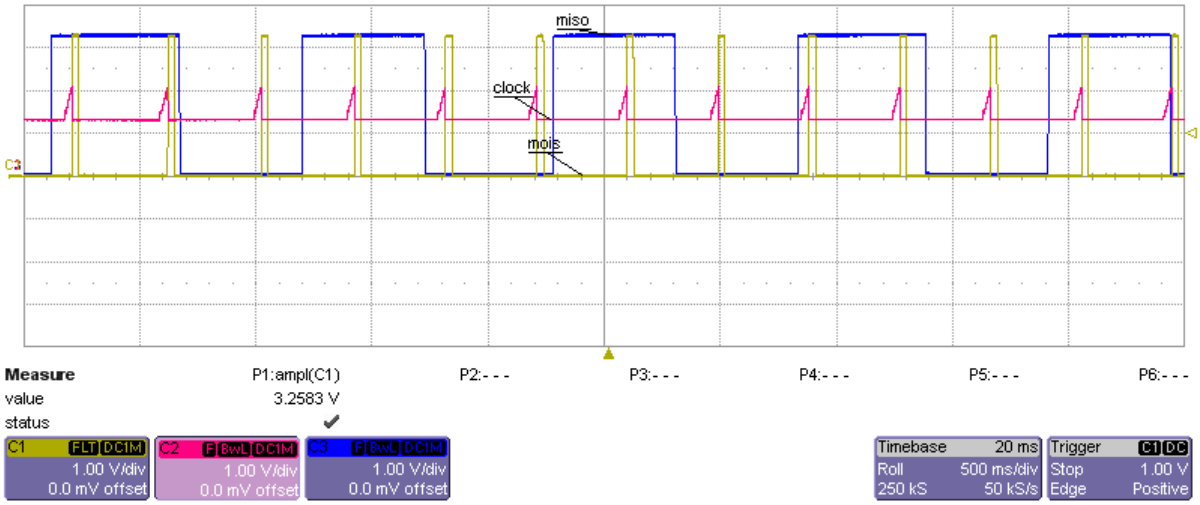


Figure 6.11: Continuous data acquisition from polling the sensor array (red trace) with simultaneous magnetic field generation (blue trace). The CTS (yellow trace) handshake signal is used to control the flow of data.

Due to limitations on the PIC18LF4680 micro-controller, interrupts must be disabled during serial communication. In order to send acquired data to the PC, the RAM rolling buffer clock is disabled temporarily for the time taken to push all data bits onto the PC serial buffer. This introduces timing inconsistencies when cycling through the RAM buffer, as the duty cycle changes in response to a disabled interrupt. However, high clock frequencies are not needed to manipulate magnetic beads, thus it is possible to clock at a lower frequency which reduces the duty cycle percentage error.

6.5 Chemical Sensor Calibration

The calibration algorithm described in Section 5.6 was implemented in MATLAB. The algorithm is dependent on two user defined parameters: step ρ_k along the research direction and mean squared error threshold ϵ. Since the change in output voltage is scaled by the chemical gain, ρ_k was arbitrarily set to $\rho_k = \frac{1}{G_{chem}}$. The inverse relationship is practical because a small change in control gate voltage will lead to a larger change in output voltage for a higher gain.

To determine the optimal error threshold ϵ, a plot of the mean squared error against the number of iterations was obtained (Figure 6.12). It was observed that the minimum error increases with increasing chemical gain. This is due to the scaling of finite DAC resolution by a factor which is equal to the chemical gain. A gain of 10 results in a resolution of 32mV, and a minimum mean squared error of 1×10^{-4}, which is 10 times the minimum error when $G_{chem} = 1$. Hence ϵ is chosen to be a function of G_{chem} where $\epsilon = G_{chem} \times 10^{-5}$.

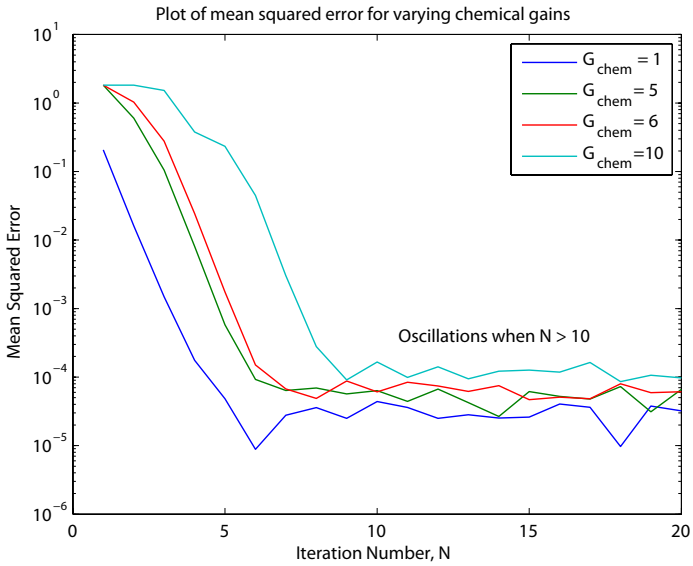


Figure 6.12: Plot of mean squared error against iteration number

The calibration algorithm was used to bias the PG-ISFET array at mid-supply (1.65V) using pH 7 buffer solution as shown in Figure 6.13. During calibration, it was also observed that the top row of ISFETs required a larger control gate bias. This could be due to varying reasons such as a low density of circuits within the area or crosstalk.. Setting parameters ρ_k and ϵ as a function of G_{chem} allows the algorithm to adapt and efficiently bias each PG-ISFET at a common operating point.

To investigate the accuracy of the calibration algorithm, the lab-on-chip was calibrated using pH 7 buffer solution, with a gain of 10 and varying ADC reference voltages. The pixels were biased within 45.8mV of each other on average, as shown in Table 6.1. Using a smaller reference voltage range does not improve calibration accuracy as the minimum resolution of a 10-bit DAC is 32mV.

Run	ADC +	ADC -	V_{max}/ V	V_{min}/ V	$V_{max} - V_{min}$/ mV
1	0.50	2.75	1.6699	1.6244	45.5
2	0.75	2.50	1.6635	1.6241	39.3
3	1.00	2.25	1.6757	1.6232	52.5
4	1.25	2.00	1.6642	1.6210	43.3
5	1.50	1.75	1.6728	1.6244	48.4
Average	-	-	1.6692	1.6234	45.8

Table 6.1: Maximum and minimum chemical readouts after calibration with varying ADC references.

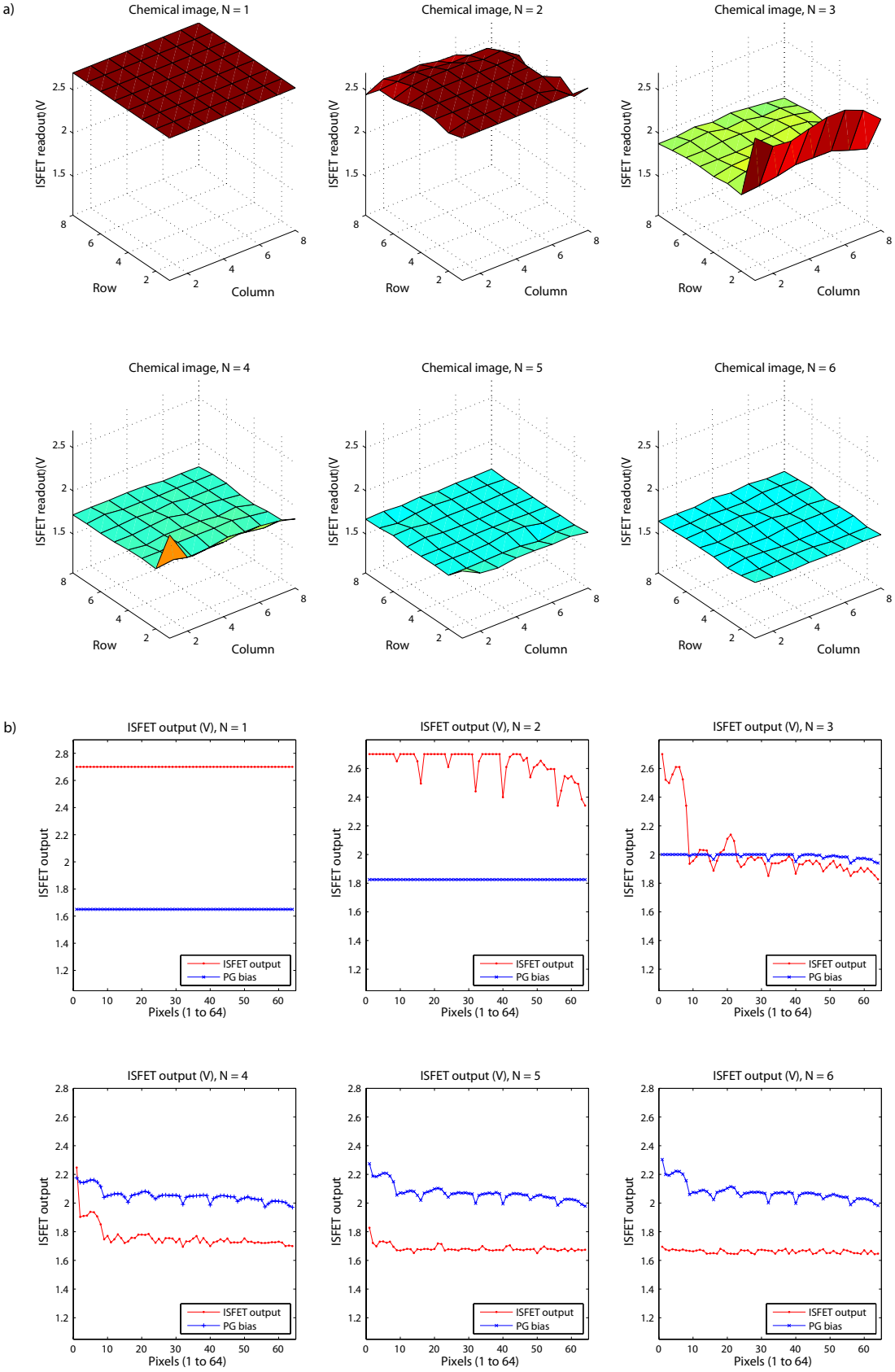


Figure 6.13: Chemical sensor calibration process using $\rho = \frac{1}{G_{chem}}$ and $\epsilon = G_{chem} \times 10^{-5}$. a) Chemical images from each iteration. b) The calibration algorithm computes the programmable gate bias (blue trace) to bias the PG-ISFET array (red trace) at mid-supply voltage (1.65V).

Chapter 7

System Characterisation

The lab-on-chip was characterised in terms of pH sensitivity, chemical drift, electrolyte potential detection and magnetic bead movement. A parylene encapsulated lab-on-chip was also characterised to investigate the effect of parylene on the lab-on-chip's sensing modalities.

In all experiments, the lab-on-chip cartridge was used to provide an elevated platform for experiments, and to prevent fluid from interfering with electronics on the development board. A signal generator was used to clock the lab-on-chip ADC and the inductive micro-coil reference current was provided by a Keithley current source. A Veho-004 USB microscope mounted on a retort stand was used to capture images. The experimental set up is shown in Figure 7.1.

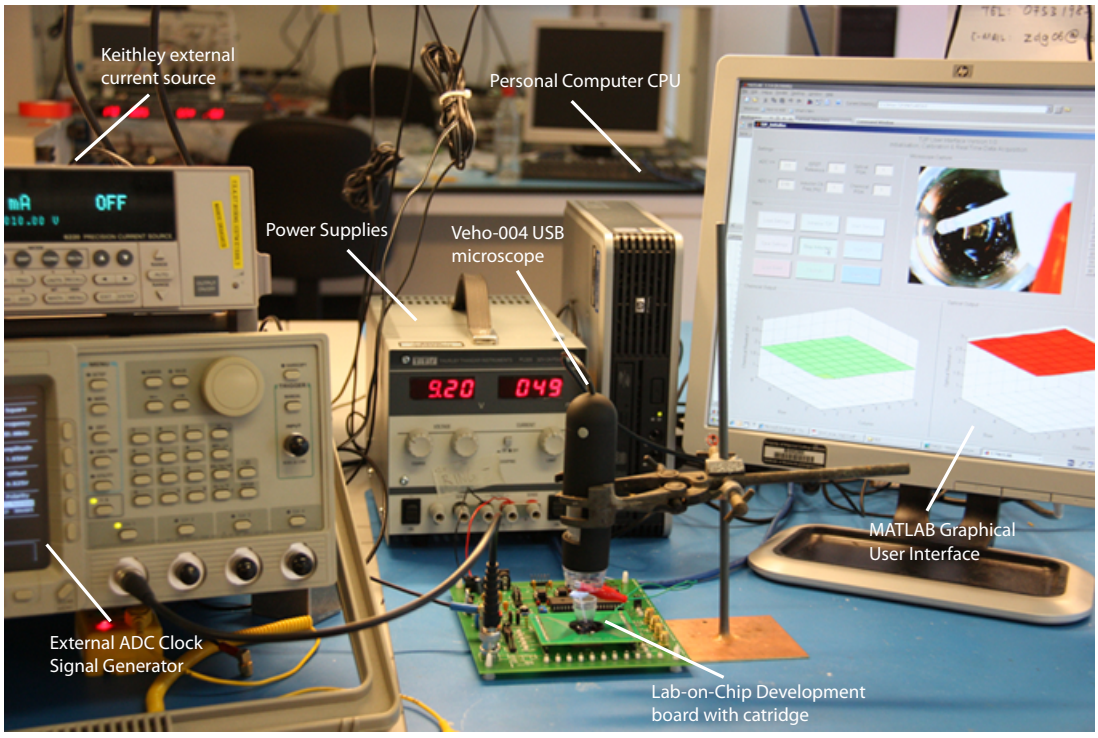


Figure 7.1: Experiment setup with external reference equipment and PC

7.1 pH-Sensitivity

The pH sensitivity of the PG-ISFET array was characterised by using solutions of varying pH. This was done by first calibrating the lab-on-chip using pH 7 buffer solution, followed by recording chemical readouts with solutions of known pH. Before a solution of different pH was used, the lab-on-chip and

fluidic well was washed with de-ionised water to remove any residual H^+ ions. In all experiments, a miniature Ag/AgCl reference electrode was used to maintain fluid potential at 0.0V and the chemical readout was allowed to settle for 2 minutes before a measurement was taken.

Figure 7.2 shows the chemical readout from all pixels and averaged chemical readouts across the PG-ISFET array for varying pH levels. The lab-on-chip was found to have a sensitivity of 57mV/pH on a gain of 10, which is 10 times less than the maximum Nernstian sensitivity of an ISFET [35]. This is due to the presence of the control gate capacitor which forms a potential divider, thus reducing the floating gate voltage as described in Appendix B.

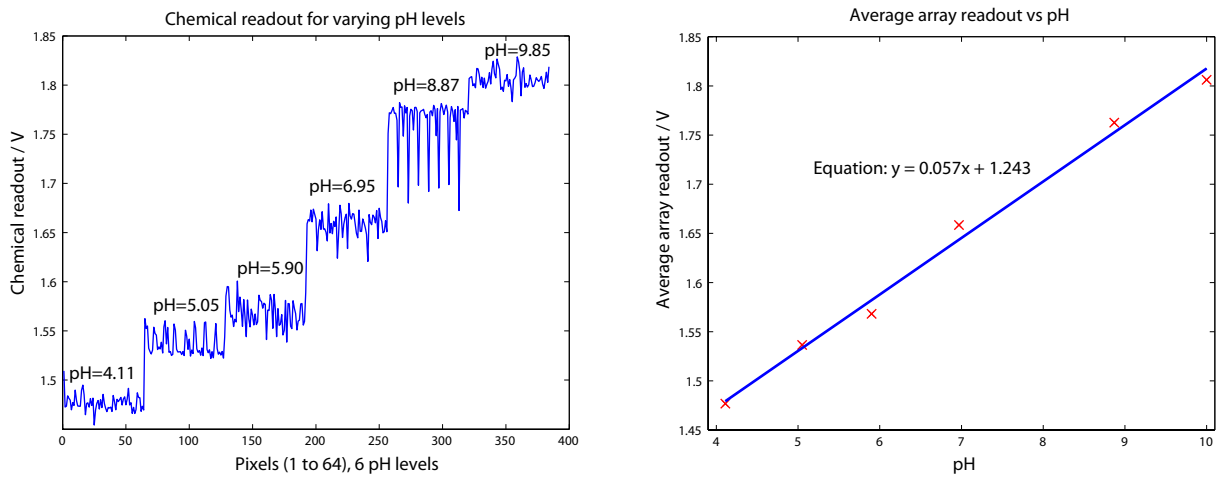


Figure 7.2: Plot of chemical readout from all 64 pixels and averaged chemical readout across the sensor array against pH.

7.2 Chemical Drift

Chemical drift was measured by first calibrating the lab-on-chip with pH 7 solution, followed by real time data acquisition with the same solution. Two successive drift tests were conducted, the first for a duration of 60 min and the second for 180 min. The lab-on-chip was re-calibrated after the first drift test, before commencement of the second 180 min drift test. The chemical readout was found to drift consistently in a decreasing manner as shown in Figure 7.3.

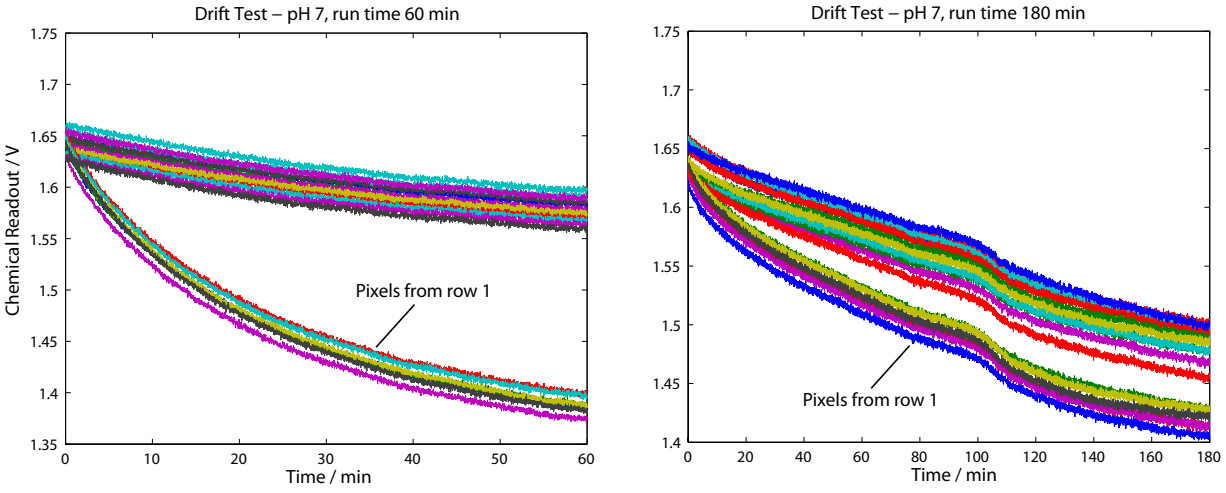


Figure 7.3: Chemical drift measured with a gain of 10 and a total time period of 60 min and 180 min.

It can be observed that the drift rate and direction for all pixels are consistent, except for row 1 pixels which drift at a faster rate than the rest of the array. Chemical drift can be modelled as slow decaying exponential with a linear approximation, where the drift rate is approximately -50mV/hr. This translates into a drift of -1pH/hr. Re-calibration of the array after the first drift test resulted in a slower drift rate for row 1 pixels: the chemical readout decreases from approximately 1.65V to 1.40V in 180 min, as compared to 1.65V to 1.40V in 60 min.

ISFET drift could be due to several reasons, including field enhanced ion migration within the gate insulator and electrochemical non-equilibrium conditions at the insulator-solution interface [36]. By modeling chemical drift over time, it is possible to implement a calibration routine which re-calibrates the array at regular intervals, hence eliminating drift.

7.3 Electrolyte Potential Detection

Changes in electrolyte potential translate into variations of the electric field across the gate insulator. This results in ISFET sensitivity to electrolyte potential, and can be exploited to detect extracellular activity of electrically active cells. The minimum detectable electrolyte potential was determined by applying a sinusoid of decreasing amplitude to the Ag/AgCl reference electrode. In all experiments, the lab-on-chip was calibrated with pH 7 buffer solution and a sinusoid of 50mHz frequency was used. Figure 7.4 shows the results of the sinusoid tests.

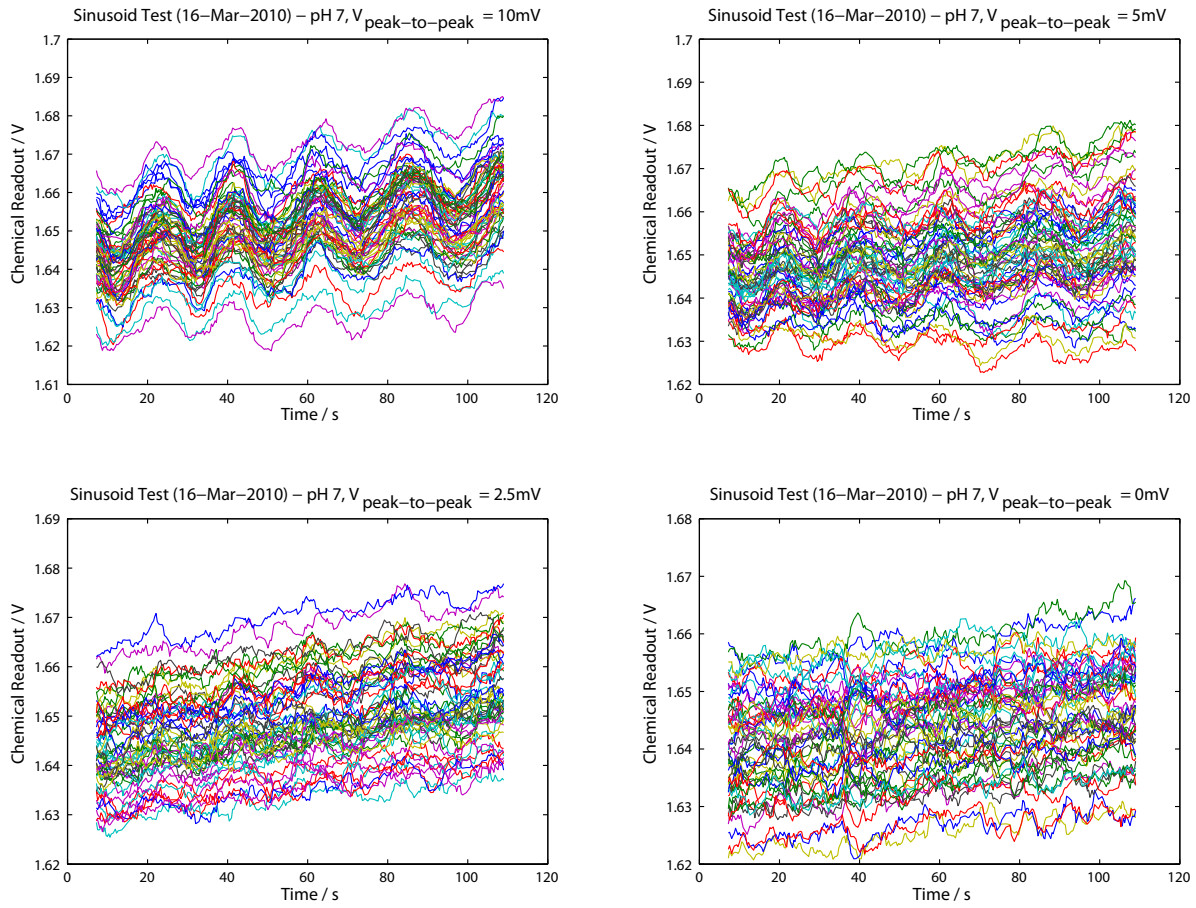


Figure 7.4: Plots of chemical readout for sinusoids of varying amplitudes applied to the reference electrode

The sinusoidal shape could not be observed for peak-to-peak amiplitudes of below 2.5mV. This is could be due to high noise levels, or solution capacitance which reduces the electrolyte potential at the chip surface. To reduce the noise levels, the signal across all pixels was averaged and is shown in Figure 7.5.

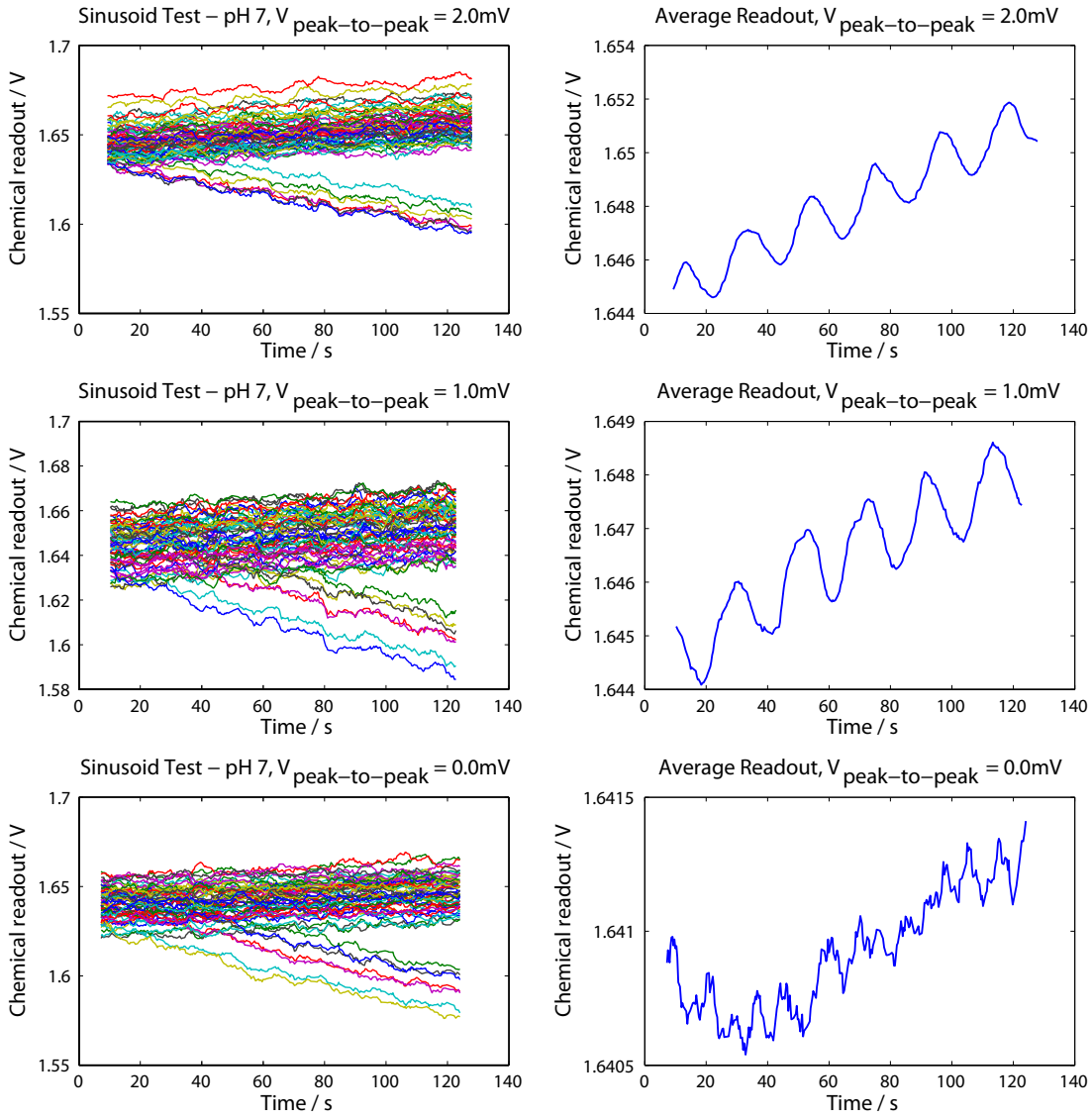


Figure 7.5: Plots of chemical readout for all pixels and average readout, for varying sinusoid amplitudes.

It is observed that averaging the chemical readout across all pixels significantly improves signal to noise ratio. The averaged chemical readout for 2mV and 1mV amplitude sinusoids clearly exhibits a sinusoid waveform. Thus it can be concluded that sinusoids of 1mV and above can be detected by the lab-on-chip by averaging across the sensor array.

The drift of row 1 pixels was found to be inconsistent with the rest of the array. This could be due to the presence of internal circuitry which influences the floating gate voltage of these pixels. In addition to this, the drift for the rest of the array follows an increasing pattern, which is opposite of that in Section 7.2. This could be due to modifications in the phenomenon of field enhanced ion migration as we go from applying a constant electrolyte potential to a sinusoidal electrolyte potential.

In terms of detecting extracellular electrical activity using individual pixels, the minimum detectable potential of 2.5mV is greater than extraceullar action potentials which lie in the range of hundreds of μV [13, 37]. However, detection of bioelectrical activity remains a possibility through the use of a sensor array. By identifying which pixels are in contact with electrically active cells, the average readout can be obtained to monitor electrical activity of neurons, cardiomyocytes and other electrically active cells.

7.4 Magnetic Bead Movement

Several magnetic field patterns were programmed to move 1.33μm diameter para-magnetic beads across the chip surface. Bead movement was not observed when biasing successive pixels on the array at various frequencies. However, when biasing individual pixels with a maximum of -21mA, bead movement in a thin film of fluid was observed as shown in Figure 7.6.

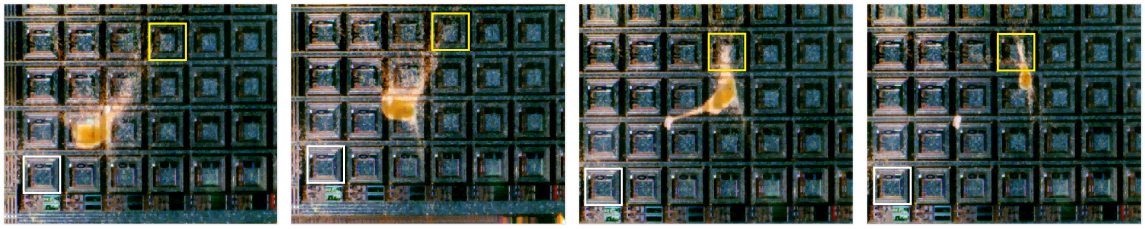


Figure 7.6: Magnetic bead movement towards the biased pixel (boxed in yellow). Pixel (8,8) is boxed in white to provide a reference point.

The montage shows that stationary beads around pixel (7,7) accumulate and move towards the pixel (5,4) which was biased with -21mA. The total time required to move beads to the biased pixel was approximately 180 seconds. Although this confirms the functionality of the inductive micro-coil array, the experiment was not readily reproducible and attempts to move beads between adjacent pixels have failed.

When compared to the results reported by Lee et. al. [8], it may be argued that the poor reproducibility could be due a large pixel area contributing to a broader magnetic field profile. Insufficient biasing current was ruled out as a possible reason as the current used by Lee et. al. was of similar magnitude [8]. Other possible reasons include surface tension of the fluid and low magnetisation of the micro-beads. It is suggested that in future designs, a smaller pixel area and a larger number of turns in each micr-coil may be able to generate sufficient magnetic force for greater manipulability.

7.5 Parylene Encapsulated Lab-on-Chip

7.5.1 pH Sensitivity

The lab-on-chip was encapsulated with parylene [38] and tested to observe the effects of encapsulation on the device. It was found that the parylene encapsulated lab-on-chip was not pH sensitive. This is due to the moisture barrier formed over the chip surface which prevents ionic interactions with the ISFET array.

Although parylene encapsulation removes pH sensitivity, it was observed that the lab-on-chip was sensitive to the presence of fluid droplets on the chip surface as shown in Figure 7.7. This is due to the change in surface capacitance which is detected by the ISFETs as a change in floating gate voltage. Hence, the system can also be modified to detect micro-droplets, in addition to its inherent chemical and optical sensing modalities.

7.5.2 Electrolyte Potential Detection

The parylene encapsulated lab-on-chip was then tested for electrolyte potential sensitivity. The lab-on-chip was first calibrated with pH 7 buffer solution and 0.0V reference electrode potential. Following this, a 100mHz sinusoid of 500mV amplitude was applied to the reference electrode. The parylene encapsulated lab-on-chip was found to be capable of detecting electrolyte potential changes, as shown in Figure 7.8.
Since the parylene encapsulated lab-on-chip devices is not ion sensitive but sensitive to electrolyte potential changes, it may be used as a reference field effect transistor (REFET). By exploiting the advantages of a sensor array, adjacent pixels can be encapsulated or ablated. Chemical readouts from parylene encapsulated pixels will contain noise only, while ablated pixels will contain both noise and pH information. This allows signal processing algorithms to be implemented for noise reduction and drift correction.

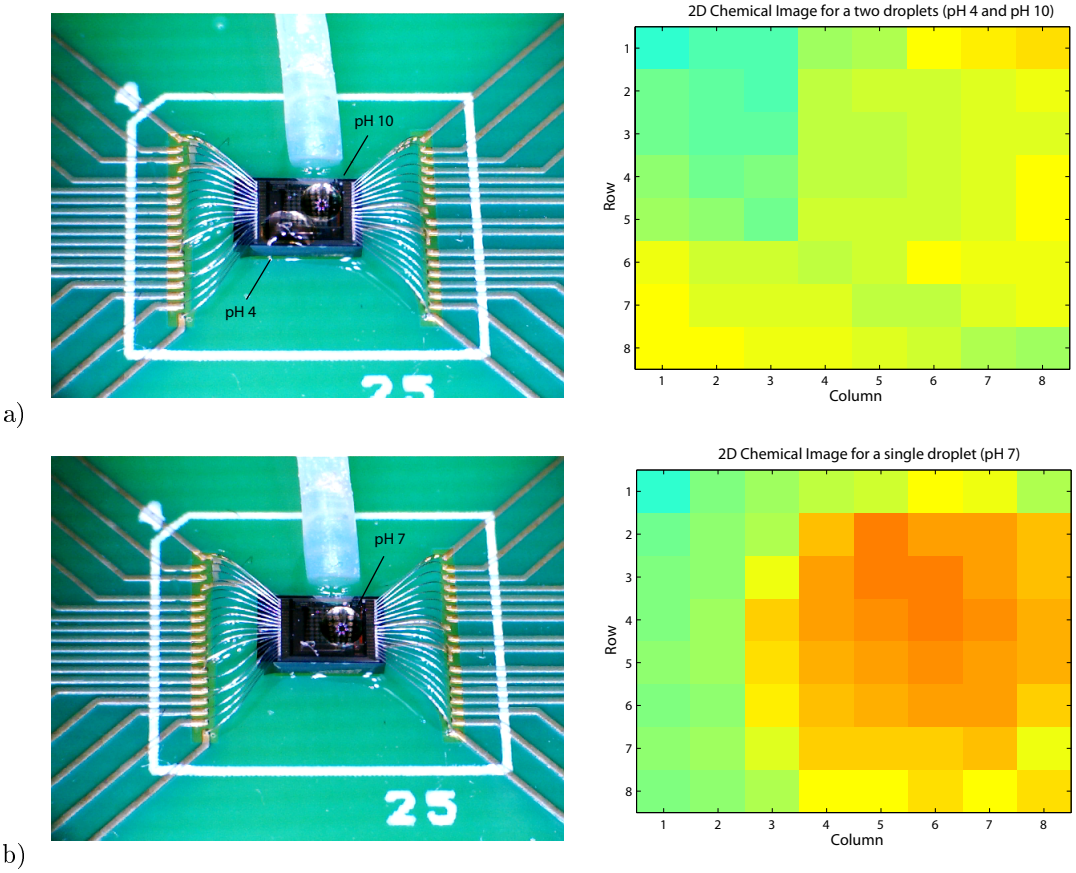


Figure 7.7: Chemical images showing the presence of fluid droplets of different pH on the chip surface. a) Two droplets of pH 4 and 10 at each corner of the chip results in the same chemical readout. b) A single droplet of pH 7 detected by the parylene encapsulated lab-on-chip.

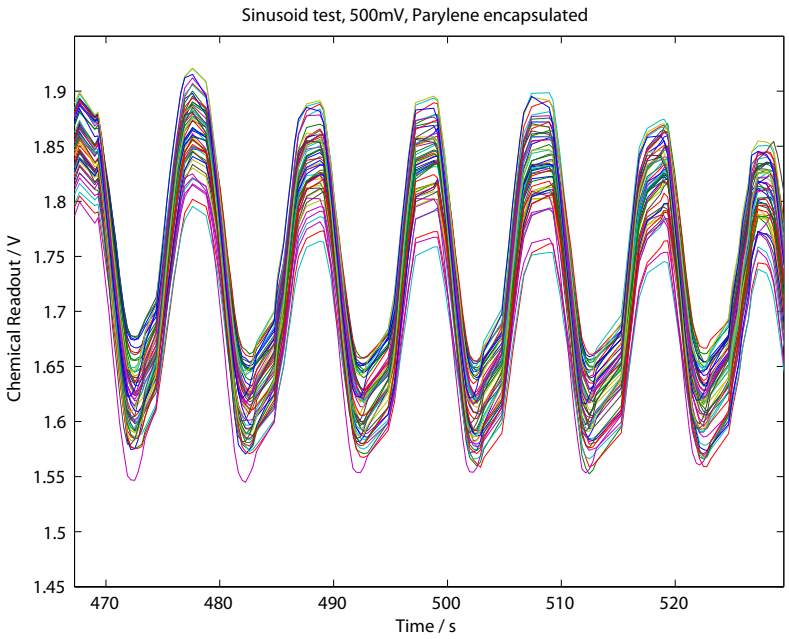


Figure 7.8: 500mV peak-to-peak sinusoid detected by the parylene encapsulated sensor array.

Chapter 8

Conclusion

This project has demonstrated the design, validation and characterisation of a novel CMOS lab-on-chip for combined magnetic manipulation and opto-chemical sensing. It is the first lab-on-chip device which integrates multiple sensing and actuation modalities on a single platform. The system has been scaled to form an 8×8 pixel array and demonstrated high pH sensitivity and optical detectability in real time, as well as the generation of micro-magnetic fields for the manipulation of magnetic beads.

The use of CMOS technology in providing lab-on-chip programmability was demonstrated in sensor calibration and magnetic actuation. Accurate sensor calibration with a fast convergence rate offers great potential for eliminating device mismatch, thus improving accuracy and reliability. Calibration combined with signal averaging across the sensor array results in significant improvement of signal to noise ratio, which enhances detection resolution. The calibration capability also opens up new opportunities in designing a programmable routine for drift compensation.

The use of the lab-on-chip as a REFET was demonstrated by encapsulating the device with parylene. By encapsulating and ablating selected pixels on the sensor array, noise cancellation and drift correction algorithms may be implemented within a single system in future designs. The use of parylene encapsulation can also be used to transform the lab-on-chip into a fluid detector which can be used to detect the presence and translocation of micro-droplets across the sensor array.

The motion of magnetic micro-beads was controlled by user configurable magnetic field patterns, transforming the lab-on-chip into a universal sensing and micro-actuation platform. Despite poor reproducibility, movement of magnetic beads towards selectively biased pixels was achieved. Fabricated in unmodified CMOS the chip can be combined with various microfluidic modalities. Fabricated in unmodified CMOS, the system can be used in multiple levels of functional integration to study unprecedented applications in cell and molecular biology.

Chapter 9

Future Work

9.1 Chemical Drift Compensation

The use of a calibration algorithm enabled the array of PG-ISFETs to be biased at a fixed operating point. By modelling chemical drift, a calibration routine can be implemented to adjust control gate voltage levels over time. This allows for chemical drift compensation and a more reliable chemical readout as simulated in Figure 9.1.

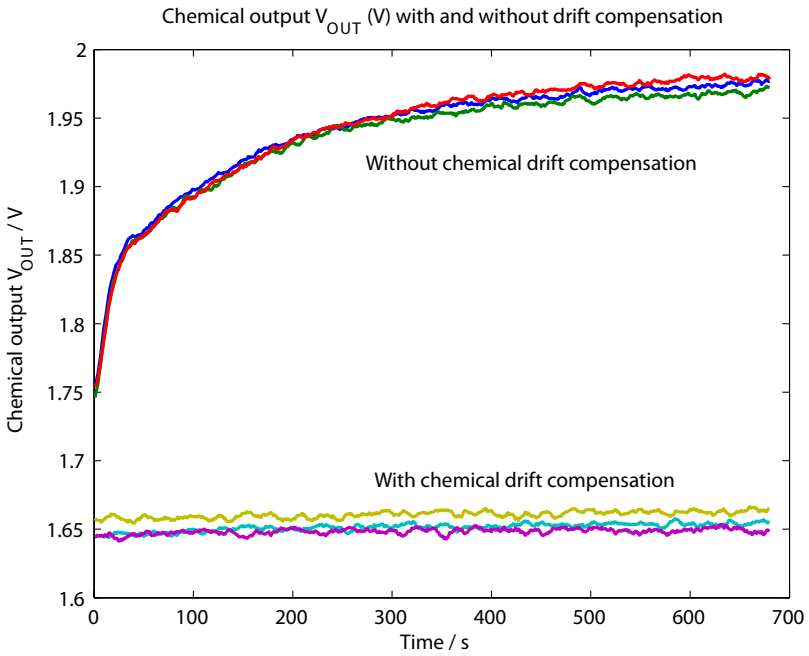


Figure 9.1: Simulated chemical readout with chemical drift compensation

9.2 Noise Reduction

Due to the small dynamic range sensor readout, electronic noise has a detrimental effect on signal to noise ratio. By taking advantage of CMOS technology, digital filters can be implemented within a single system to reduce sensor readout noise. In addition to this, the sensor array can be encapsulated with parylene, with every other pixel being ablated to regain pH sensitivity. Each pair of correlated pixel readouts consists of a signal containing noise only, and another containing both signal and noise. Subtraction of both signals can lead to significant noise reduction and improved signal to noise ratios.

9.3 Characterising Optical Sensitivity

Bioluminescence and electrochemiluminescence are the most common luminometric bioassays which employ the use of a photodiode array. To accurately measure luminescence levels from the optical readout, the lab-on-chip must be characterised in terms of optical sensitivity. As the optical readout depends on the time taken to discharge the photodiode, it is also essential to determine the optimum frequency at which the sensors are polled. The use of a low frequency clock will result in higher optical sensitivity but a faster rate of saturation, while a high frequency clock will lead to a larger detection range but lower optical sensitivity.

9.4 Characterising Crosstalk

System characterisation was performed by polling the sensor array and generating magnetic field patterns independently. To understand the effects of crosstalk, the system must be characterised with multiple modalities running simultaneously. Crosstalk may arise from capacitative, inductive or conductive coupling from one part of the system to another. The characterisation process must take into consideration the effect of illumination intensity on chemical sensing and drift, the effect of inductive coil heating on sensor readout and the effect of varying clock frequencies on sensor readout.

9.5 Characterising Magnetic Bead Movement

The movement of magnetic beads to a single biased pixel was demonstrated by the inductive microcoil array. However, the system failed to achieve precise translocation of magnetic beads across various locations on the lab-on-chip. This may be achieved by reducing the pixel size by using 0.18μm CMOS technology, thus increasing the localisation of the micro-magnetic field. The system should also be characterised in terms of maximum manipulable bead size and minimum biasing current. Magnetic beads must also be tagged with cells to observe the effect of cell tagging on manipulability

9.6 Microfluidic Integration

Microfluidic channels can be integrated with the lab-on-chip for applications which require channelling fluid flow, such as fluorescence activated cell sorting. An experimental microfluidic platform shown in Figure 9.2 can be used to test the lab-on-chip in various applications such as measuring flow velocity and microparticle photometry. The microfluidic dies will be fabricated based on Polydimethylsiloxane (PDMS) membranes, which are bonded to glass slides with an inlet and outlet port. The channels are photolithographically patterned on a negative photoresist (SU-8) layer, and spin-coated on a silicon wafer which serves as the mold for PDMS casting.

9.7 Application to Biochemistry and Cell Biology

The full functionality of the lab-on-chip is realised in various micro-scale applications. At the lowest functional level, a single modality can be used to record electrical signals from a nerve fibre (Figure 9.3) or to detect luminescence in bioluminometric assays. At the next functional level, any two modalities are used simultaneously in applications such as micro-particle photometry and bioluminometric SNP typing assays with nucleotide immobilisation. At the top level, all three modalities are employed in novel applications such as monitoring the effect of magnetic bead drug delivery to biological cells and magnetically controlled fluorescence activated cell sorting.

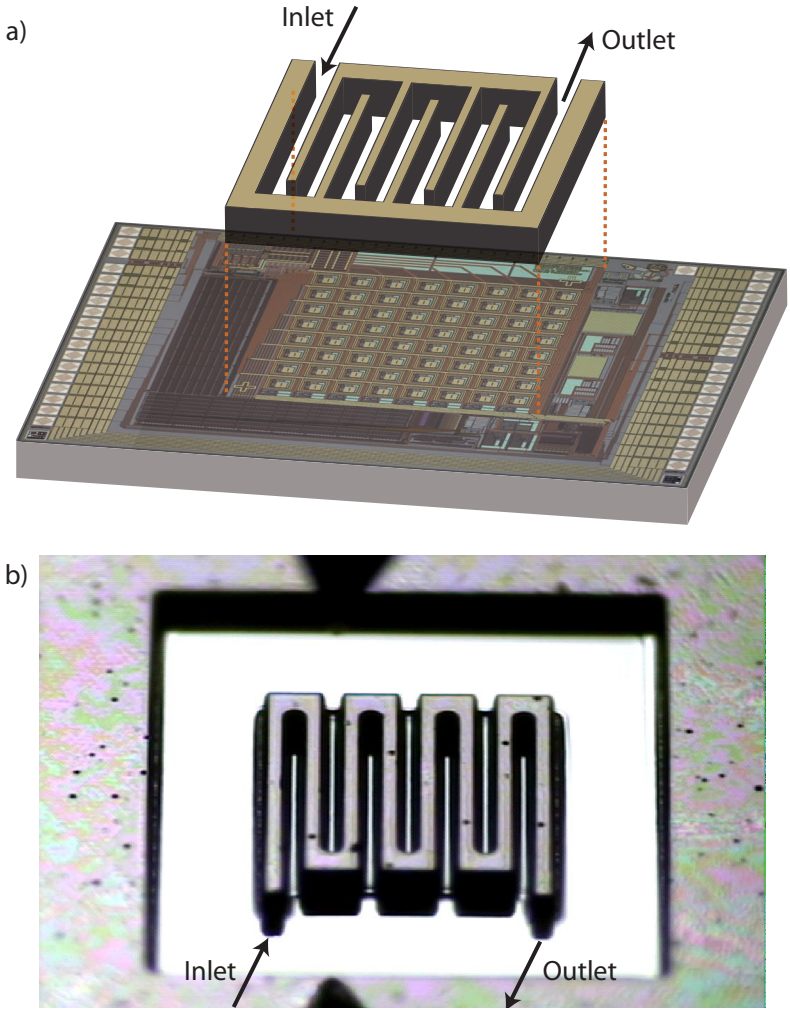


Figure 9.2: Integrated microfluidic platform with the lab-on-chip. a) Illustration of the microfluidic platform with a meandering channel on the chip surface [7]. b) Microphotograph of the fabricated microfluidic channels.

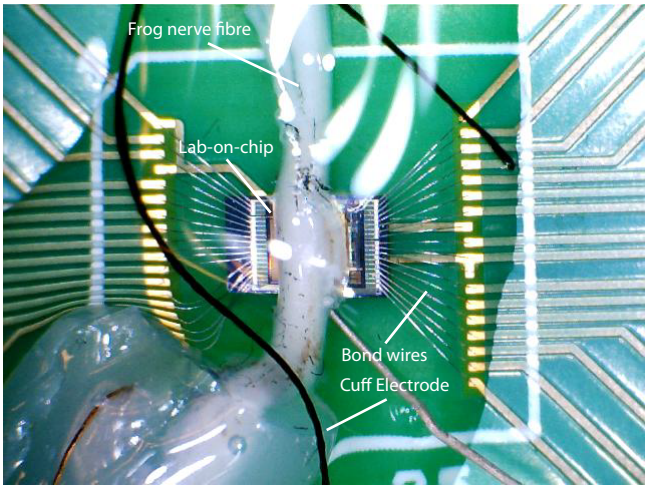


Figure 9.3: Recording electrical activity from an isolated frog nerve fibre.

Bibliography

[1] Helene Andersson and Albert van den Berg. Microfluidic devices for cellomics: a review. *Sensors and Actuators B: Chemical*, 92(3):315 – 325, 2003.

[2] Mengsu Yang, Cheuk-Wing Li, and Jun Yang. Cell docking and on-chip monitoring of cellular reactions with a controlled concentration gradient on a microfluidic device. *Analytical Chemistry*, 74(16):3991–4001, August 2002.

[3] Jin-Woo Choi, Chong H. Ahn, Shekhar Bhansali, and H. Thurman Henderson. A new magnetic bead-based, filterless bio-separator with planar electromagnet surfaces for integrated bio-detection systems. *Sensors and Actuators B: Chemical*, 68(1-3):34 – 39, 2000.

[4] Leandro Lorenzelli, Benno Margesin, Sergio Martinoia, M. T. Tedesco, and Maurizio Valle. Bio-electrochemical signal monitoring of in-vitro cultured cells by means of an automated microsystem based on solid state sensor-array. *Biosensors and Bioelectronics*, 18(5-6):621 – 626, 2003. Selected papers from the Seventh World Congress on Biosensors Kyoto, Japan 15-17 May 2002.

[5] Masao Kamahori, Kunio Harada, and Hideki Kambara. A new single nucleotide polymorphisms typing method and device by bioluminometric assay coupled with a photodiode array. *Measurement Science and Technology*, 13(11):1779–1785, 2002.

[6] T. C. W. Yeow, M. R. Haskard, D. E. Mulcahy, H. I. Seo, and D. H. Kwon. A very large integrated ph-isfet sensor array chip compatible with standard cmos processes. *Sensors and Actuators B: Chemical*, 44(1-3):434 – 440, 1997.

[7] P. Georgiou, T. Constandinou, T. Prodomakis, and C. Toumazou. A cmos-based lab-on-chip array for the combined magnetic stimulation and opto-chemical sensing of neural tissue. In *Cellular Nanoscale Networks and Applications*, 2010.

[8] H Lee. Integrated cell manipulation system - cmos/microfluidic hybrid. *Lab on a chip*, 7(3):331–, 2007.

[9] H. Lee, A. M. Purdon, and R. M. Westervelt. Manipulation of biological cells using a microelectromagnet matrix. *Applied Physics Letters*, 85(6):1063–1065, 2004.

[10] G E Moore. Cramming more components onto integrated circuits. *Proceedings of the IEEE*, 86(1):82–, 1998.

[11] R R Schaller. Moore's law: past, present and future. *IEEE spectrum*, 34(6):52–, 1997.

[12] T G Constandinou. A micropower centroiding vision processor. *IEEE journal of solid-state circuits*, 41(6):1430–, 2006.

[13] B Eversmann. A 128 x 128 cmos biosensor array for extracellular recording of neural activity. *IEEE journal of solid-state circuits*, 38(12):2306–, 2003.

[14] H Eltoukhy. A 0.18um cmos bioluminescence detection lab-on-chip. *IEEE journal of solid-state circuits*, 41(3):651–, 2006.

[15] Sergio Martinoia, Nicola Rosso, Massimo Grattarola, Leandro Lorenzelli, Benno Margesin, and Mario Zen. Development of isfet array-based microsystems for bioelectrochemical measurements of cell populations. *Biosensors and Bioelectronics*, 16(9-12):1043 – 1050, 2001.

[16] Arshak Poghossian and Michael J. Schöning. Detecting both physical and (bio-)chemical parameters by means of isfet devices. *Electroanalysis*, 16(22):1863–1872, 2004.

[17] Arshak Poghossian and Michael J. Schöning. Recent advances in biologically sensitive field-effect transistors (biofets). *The Analyst*, 127(9):1137–1151, 2002.

[18] E. Souteyrand, J. P. Cloarec, J. R. Martin, C. Wilson, I. Lawrence, S. Mikkelsen, and M. F. Lawrence. Direct detection of the hybridization of synthetic homo-oligomer dna sequences by field effect. *The Journal of Physical Chemistry B*, 101(15):2980–2985, April 1997.

[19] E.R. Fossum. Cmos image sensors: electronic camera on a chip. In *Electron Devices Meeting, 1995., International*, pages 17–25, Dec 1995.

[20] Jonathan K. Leland and Michael J. Powell. Electrogenerated chemiluminescence: An oxidative-reduction type ecl reaction sequence using tripropyl amine. *Journal of The Electrochemical Society*, 137(10):3127–3131, 1990.

[21] Sukdeb Pal, Min Kim, Yu Tak, Ho Kwon, and Joon Song. Interdigitated microelectrode array-coupled bipolar semiconductor photodiode array (imea-pda) microchip for on-chip electrochemiluminescence detection. *Biomedical Microdevices*, 11(5):971–980, October 2009.

[22] Jeong-Gun Lee, Kyusik Yun, Guei-Sam Lim, Sang Eun Lee, Suhyeon Kim, and Je-Kyun Park. Dna biosensor based on the electrochemiluminescence of ru(bpy)32+ with dna-binding intercalators. *Bioelectrochemistry*, 70(2):228 – 234, 2007.

[23] J. Kruger, K. Singh, A. O'Neill, C. Jackson, A. Morrison, and P. O'Brien. Development of a microfluidic device for fluorescence activated cell sorting. *Journal of Micromechanics and Microengineering*, 12(4):486–494, 2002.

[24] Q A Pankhurst. Applications of magnetic nanoparticles in biomedicine. *Journal of physics. D, Applied physics*, 36(13):167–, 2003.

[25] S Morisada. Immunomagnetic separation of scum-forming bacteria using polyclonal antibody that recognizes mycolic acids. *Journal of microbiological methods*, 51(2):141–, 2002.

[26] U Lehmann. Microparticle photometry in a cmos microsystem combining magnetic actuation and in situ optical detection. *Sensors and actuators. B, Chemical*, 132(2):411–, 2008.

[27] Leila Shepherd and Chris Toumazou. Weak inversion isfets for ultra-low power biochemical sensing and real-time analysis. *Sensors and Actuators B: Chemical*, 107(1):468 – 473, 2005. Proceedings of the 7th European Conference on Optical Chemical Sensors and Biosensors - EUROPT(R)ODE VII.

[28] P. Georgiou and C. Toumazou. An adaptive cmos-based pg-isfet for ph sensing. In *Circuits and Systems, 2009. ISCAS 2009. IEEE International Symposium*, pages 557–560, May 2009.

[29] P.A. Hammond, D. Ali, and D.R.S. Cumming. Design of a single-chip ph sensor using a conventional 0.6-um cmos process. *Sensors Journal, IEEE*, 4:706–712, 2004.

[30] M.J. Milgrew and D.R.S. Cumming. Matching the transconductance characteristics of cmos isfet arrays by removing trapped charge. *Electron Devices, IEEE Transactions on*, 55(4):1074–1079, April 2008.

[31] R.H. Dyck and G.P. Weckler. Integrated arrays of silicon photodetectors for image sensing. *Electron Devices, IEEE Transactions on*, 15(4):196–201, Apr 1968.

[32] Tom Greig, Chris Castelli, Andrew Holland, and David Burt. The design of an active pixel sensor test structure optimised for the read out of scintillator screens. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 573(1-2):30 – 33, 2007. Proceedings of the 7th International Conference on Position-Sensitive Detectors - PSD-7, 7th International Conference on Position-Sensitive Detectors.

[33] N. Wongkomet and B.E. Boser. Correlated double sampling in capacitive position sensing circuits for micromachined applications. In *Circuits and Systems, 1998. IEEE APCCAS 1998. The 1998 IEEE Asia-Pacific Conference on*, pages 723–726, Nov 1998.

[34] R.J. Van De Plassche and H.J. Schouwenaars. A monolithic 14 bit a/d converter. *Solid-State Circuits, IEEE Journal of*, 17(6):1112–1117, Dec 1982.

[35] P. Bergveld. Thirty years of isfetology: What happened in the past 30 years and what may happen in the next 30 years. *Sensors and Actuators B: Chemical*, 88(1):1 – 20, 2003.

[36] Masayoshi Esashi and Tadayuki Matsuo. Integrated micro multi ion sensor using field effect of semiconductor. *Biomedical Engineering, IEEE Transactions on*, BME-25(2):184 –192, march 1978.

[37] Ariel Cohen, Micha E. Spira, Shlomo Yitshaik, Gustaaf Borghs, Ofer Shwartzglass, and Joseph Shappir. Depletion type floating gate p-channel mos transistor for recording action potentials generated by cultured neurons. *Biosensors and Bioelectronics*, 19(12):1703 – 1709, 2004.

[38] Prodromakis T, Michelakis K, Zoumpoulidis T, Dekker R, and Toumazou C. Biocompatible encapsulation of cmos based chemical sensors. In *IEEE Sensors Conference*, pages 791–794, 2009.

[39] Martin Gijs. Magnetic bead handling on-chip: new opportunities for analytical applications. *Microfluidics and Nanofluidics*, 1:22–40, 2004.

Appendix A

Magnetic Force on a Super-paramagnetic Micro-bead

The magnetic force acting on a magnetic moment $\boldsymbol{m}$ is given by:

$$\boldsymbol{F_m} = \frac{1}{\mu_0}\boldsymbol{\nabla}\left(\boldsymbol{m}\cdot\boldsymbol{B}\right) \approx \frac{1}{\mu_0}\left(\boldsymbol{m}\cdot\boldsymbol{\nabla}\right)\boldsymbol{B} \tag{A.1}$$

The approximation is true, assuming that the magnetic moment does not vary with space, i.e. $(\boldsymbol{\nabla}\cdot\boldsymbol{m}=0)$. This assumption only holds if the magnetic moment is permanent or when the magnetic particle is in a sufficiently large magnetic field such that its magnetisation is completely saturated [39].

When considering super-paramagnetic micro-beads suspended in a weakly diamagnetic medium, the total magnetic moment $\boldsymbol{m}$ on the particle can be re-written as:

$$\boldsymbol{m} = V_m\boldsymbol{M} \tag{A.2}$$

where V_m is the volume of the particle and $\boldsymbol{M}$ is the net volumetric magnetisation, given by

$$\boldsymbol{M} = \triangle\chi\boldsymbol{H} \tag{A.3}$$

where $\triangle\chi = \chi_m - \chi_w$ is the effective susceptibility of the particle relative to water and $\boldsymbol{H}$ is the magnetic field strength. Since the magnetic flux density is related to the magnetic field strength by $\boldsymbol{B} = \mu_0\boldsymbol{H}$, we substitute Equations A.2 and A.3 into EquationA.1 and obtain:

$$\boldsymbol{F_m} = \frac{V\triangle\chi}{\mu_0}\left(\boldsymbol{B}\cdot\boldsymbol{\nabla}\right)\boldsymbol{B} \tag{A.4}$$

Assuming that there are no time-varying electric fields or currents in the medium, we can apply Maxwell's equation $\boldsymbol{\nabla}\times\boldsymbol{B}=0$ to the identity:

$$\boldsymbol{\nabla}\left(\boldsymbol{B}\cdot\boldsymbol{B}\right) = 2\boldsymbol{B}\times\left(\boldsymbol{\nabla}\times\boldsymbol{B}\right) + 2\left(\boldsymbol{B}\cdot\boldsymbol{\nabla}\right)\boldsymbol{B} = 2\left(\boldsymbol{B}\cdot\boldsymbol{\nabla}\boldsymbol{B}\right) \tag{A.5}$$

Substituting Equation A.5 into Equation A.4, we obtain the expression

$$\boldsymbol{F_m} = \frac{V\triangle\chi}{2\mu_0}\boldsymbol{\nabla}\boldsymbol{B}^2 = V\triangle\chi\boldsymbol{\nabla}\left(\frac{1}{2}\boldsymbol{B}\cdot\boldsymbol{H}\right) \tag{A.6}$$

where $\boldsymbol{\nabla}\left(\frac{1}{2}\boldsymbol{B}\cdot\boldsymbol{H}\right)$ represents the differential of the magnetic field energy density [24]. This means that the magnetic force always acts in the direction of the steepest gradient of the magnetic energy density field. It also forms the basis for magnetic manipulation of micro-beads in bio-assays.

Appendix B

Operation of the PG-ISFET

The operation of an ISFET is based on threshold voltage variations due to ionic interactions at the oxide-solution interface. The threshold voltage V_t of an ISFET is given by the expression:

$$V_t = E_{ref} - \left(\Psi - \chi^{sol}\right) - \frac{\Phi_{Si}}{q} - \frac{Q_{ox} + Q_{ss} + Q_B}{C_{ox}} + 2\phi_f \tag{B.1}$$

where E_{ref} is the reference electrode potential, Ψ is the electrostatic potential, χ^{sol} is the surface dipole potential of the solution, Φ_{Si} is the silicon work function, q is the elementary charge, Q_{ox} is oxide charge, Q_{ss} is the charge at the oxide-silicon interface, Q_B is the depletion charge, C_{ox} is the oxide capacitance, and ϕ_f is the Fermi potential [35]. All terms are constant except Ψ which is a function of pH:

$$\Psi = 2.3\alpha\frac{kT}{q}\text{pH} \tag{B.2}$$

$$\alpha = \frac{1}{\left(2.3kT/q^2\right)\left(C_s/\beta_s\right) + 1} \tag{B.3}$$

where k is the Boltzmann constant, T is the absolute temperature, α is a dimensionless sensitivity constant, C_s is the differential double layer capacitance and β_s is the surface buffer capacity.

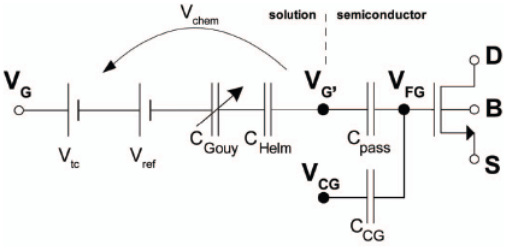


Figure B.1: Schematic of the PG-ISFET [28]

Next, we consider the effect of trapped charges on the operation of the ISFET. By modeling the ISFET based on the Gouy-Chapman-Stern double layer model [27, 28], the effective gate voltage is expressed as:

$$V_G' = V_{ref} - V_{tc} - V_{chem} \tag{B.4}$$

where V_{ref} is the reference electrode bias, V_{tc} is the potential due to trapped charge, and V_{chem} is the pH dependent potential. The floating gate voltage is now expressed as:

$$V_{FG} = \frac{V_G' C_{pass} + V_{CG} C_{CG} + V_S C_{GS} + V_D C_{GD}}{C_T} \tag{B.5}$$

where C_{pass} is the passivation capacitance, C_{CG} is the control gate capacitance and V_{CG} is the programmable gate bias. Like the MOSFET, V_S is the source voltage, C_{GS} is the gate-source capacitance, V_D is the drain voltage, C_{GD} is the gate-drain capacitance and C_T is the total capacitance of the ISFET. From Equations B.4 and B.5, we observe that the effect of V_{tc} can be nullified if $V_{CG}C_{CG} = -V_{tc}C_{pass}$. Hence, by applying a voltage bias V_{CG} at the control gate, we can eliminate sensor non-idealities due to trapped charge and bias an array of ISFETs to function at the same operating point.

Appendix C

Hardware Design Layout

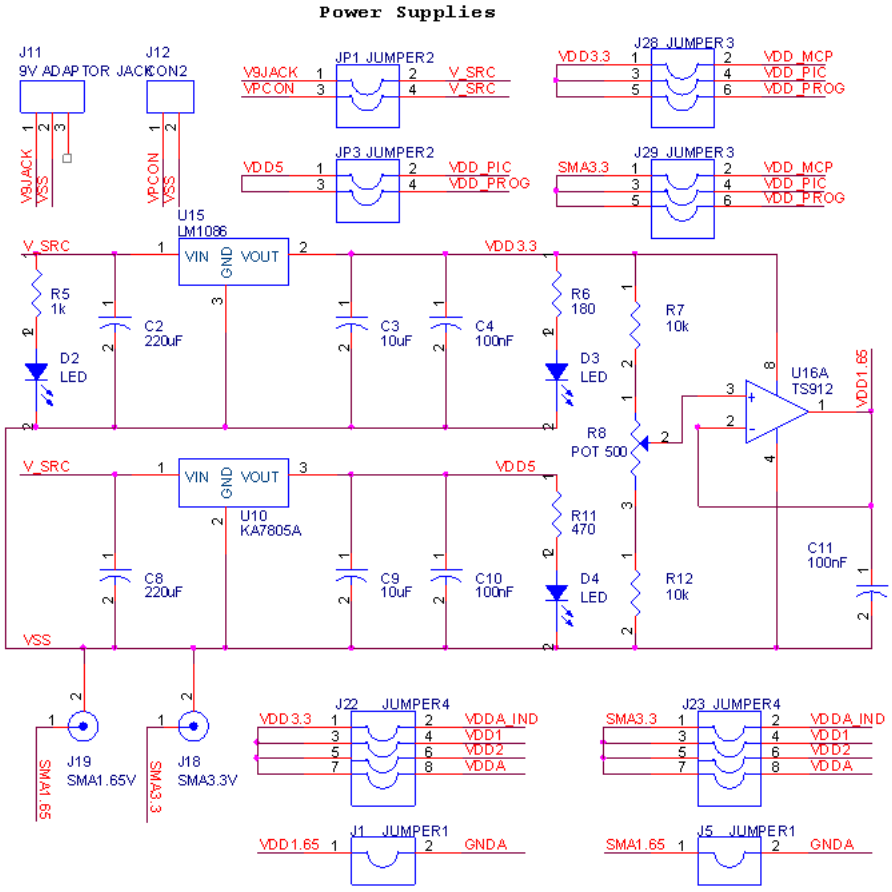


Figure C.1: Development board power supplies

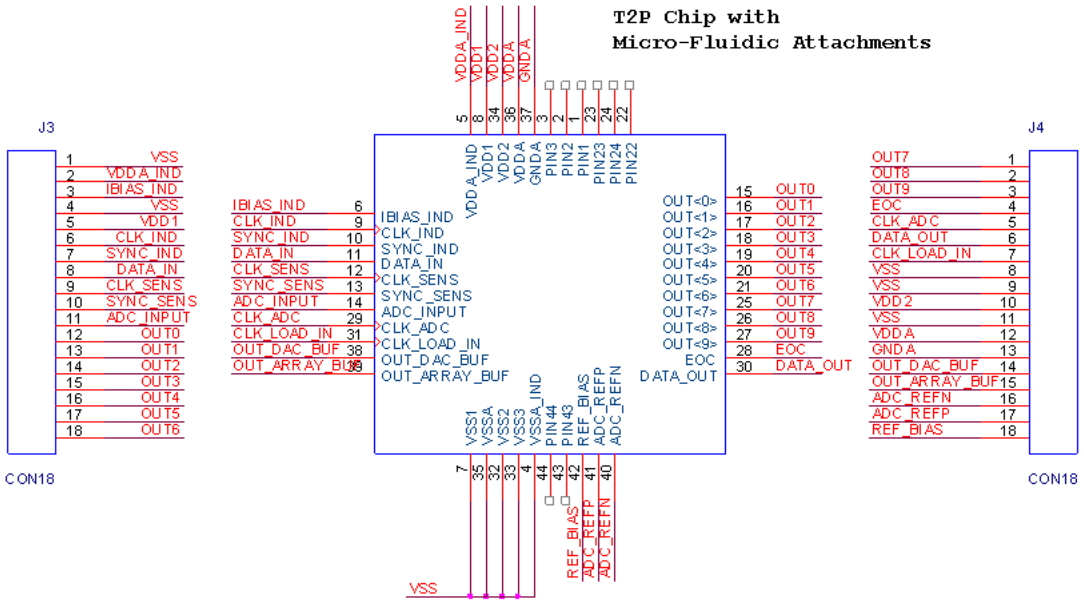


Figure C.2: Lab-on-chip device with micro-fluidic PCB connectors.

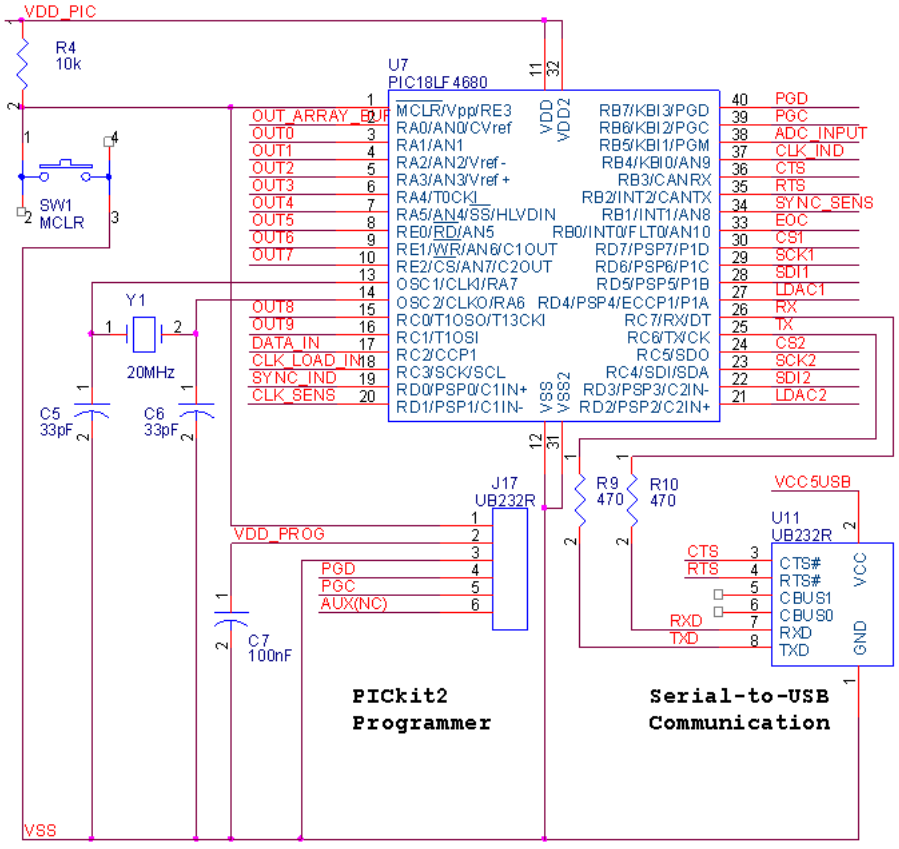


Figure C.3: PIC micro-controller with the PIC programmer and serial-to-USB communication module

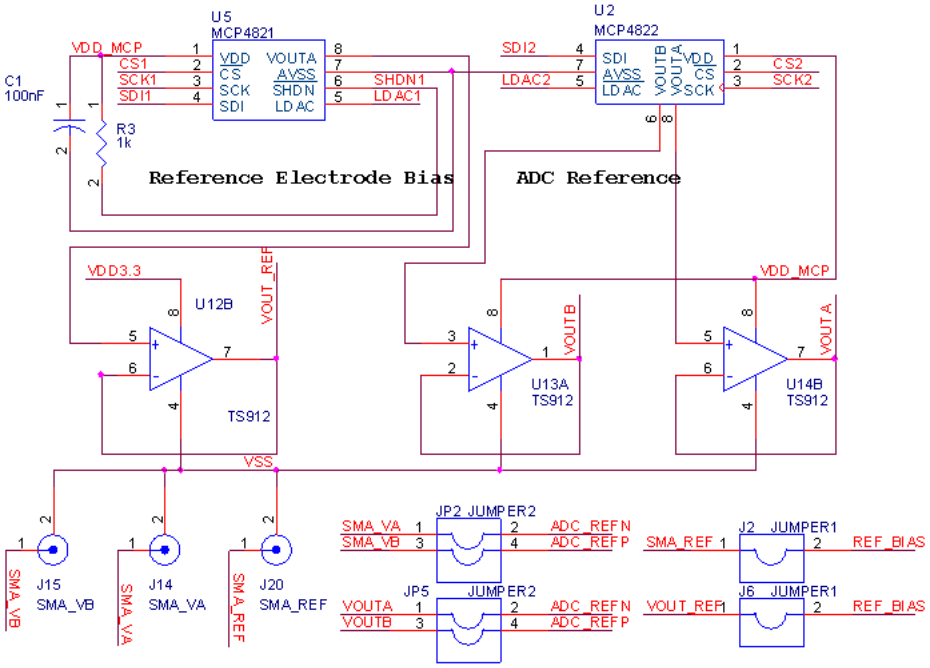


Figure C.4: Lab-on-chip reference voltages

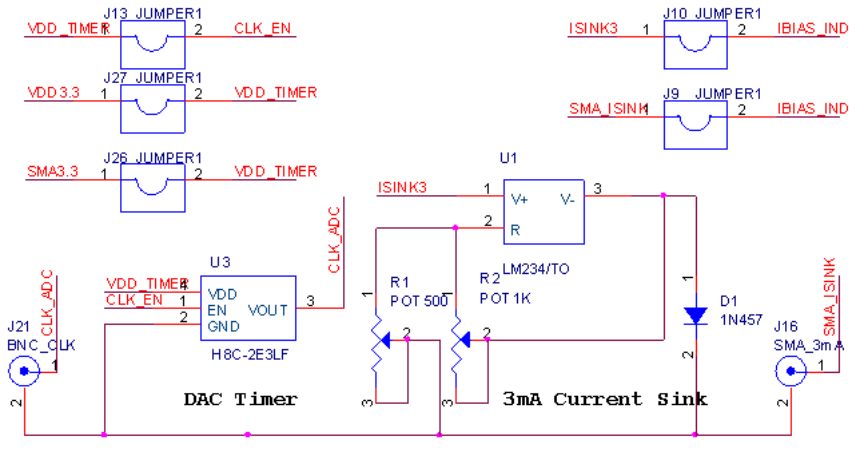
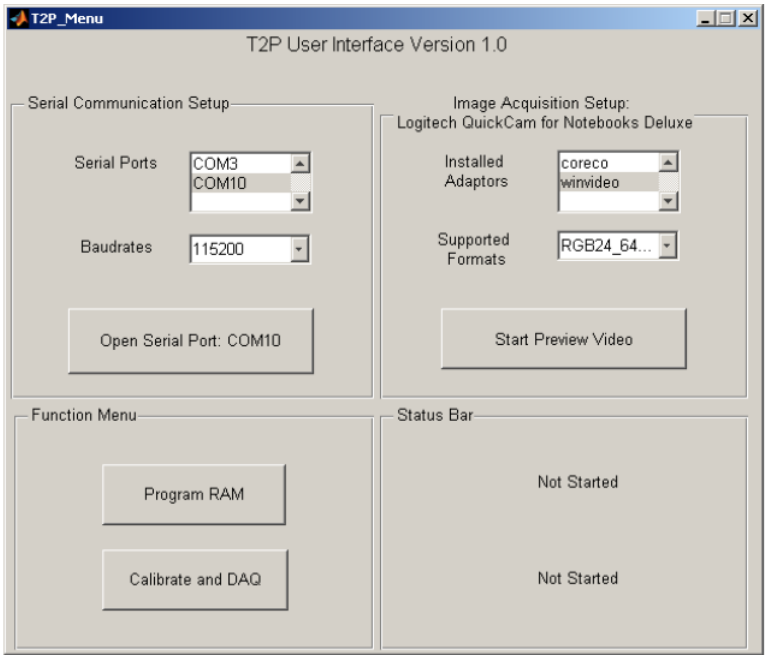


Figure C.5: DAC clock and inductive micro-coil current sink

Appendix D

MATLAB GUI Main Menu

(a)



(b)

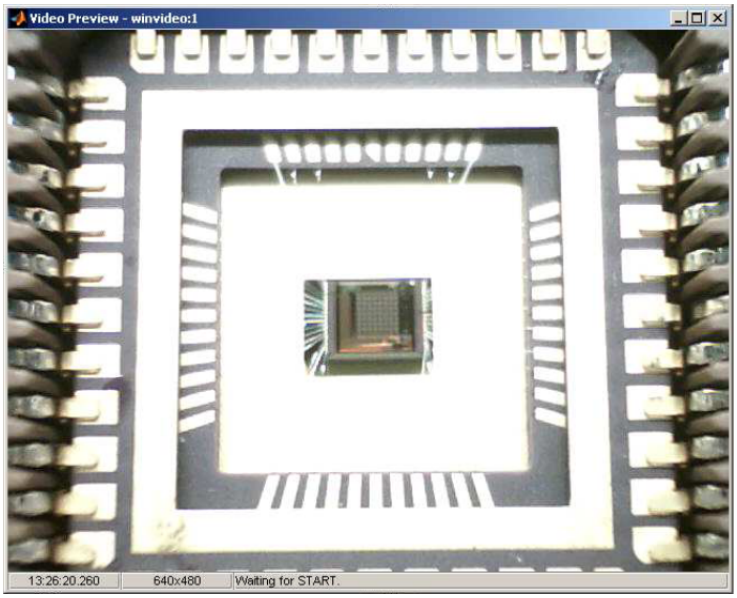


Figure D.1: MATLAB graphical user interface main menu and image acquisition preview

Appendix E

Initialisation, Calibration and Real Time Data Acquisition GUI

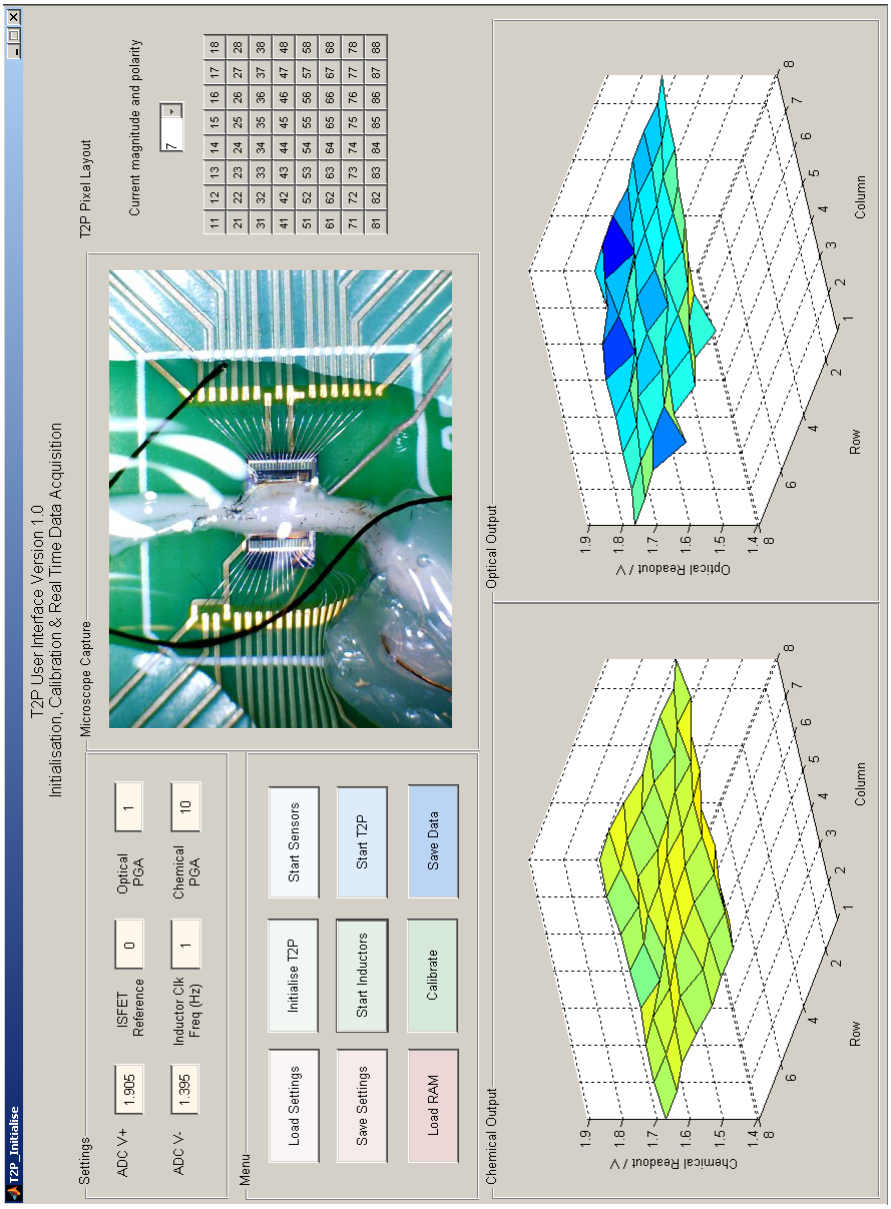


Figure E.1: Lab-on-chip initialisation, calibration and real time data acquisition GUI

Appendix F

Magnetic Field Pattern Simulation GUI

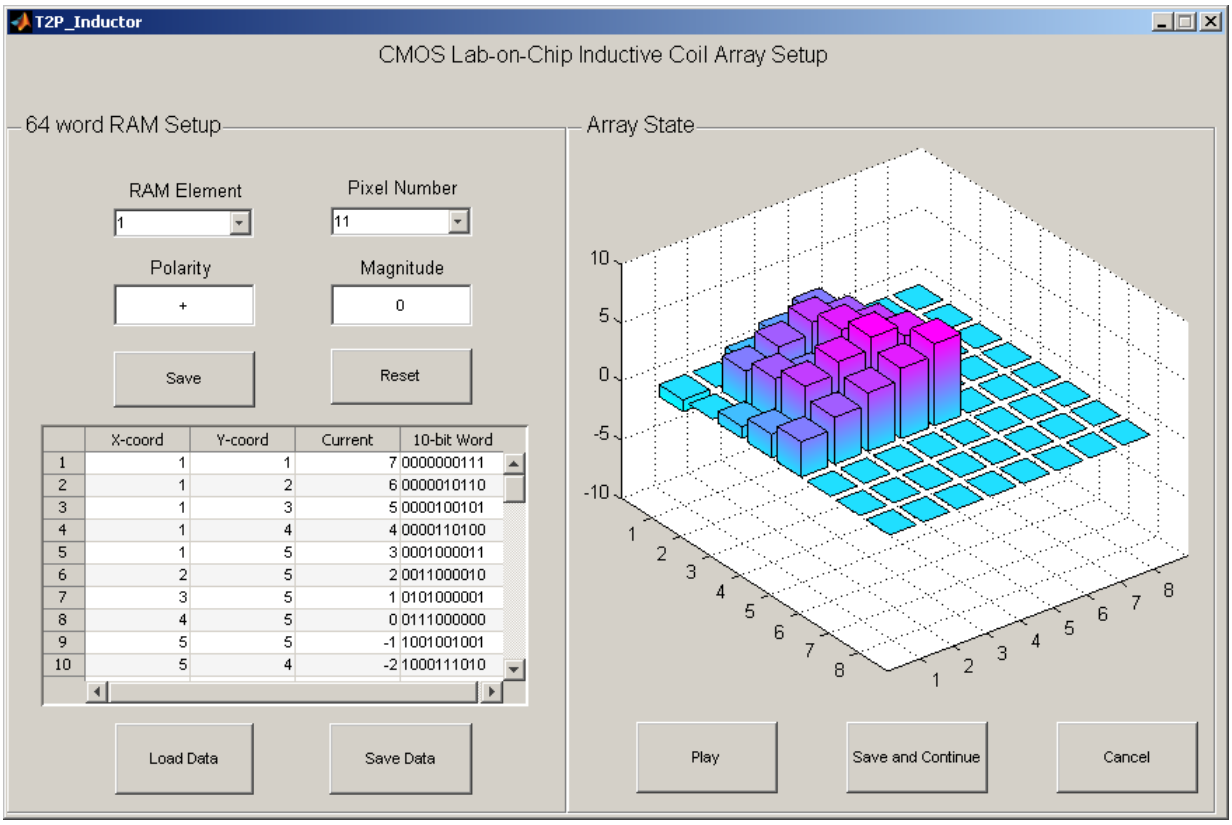


Figure F.1: Lab-on-chip magnetic field pattern simulation GUI

Appendix G

Image Analyser

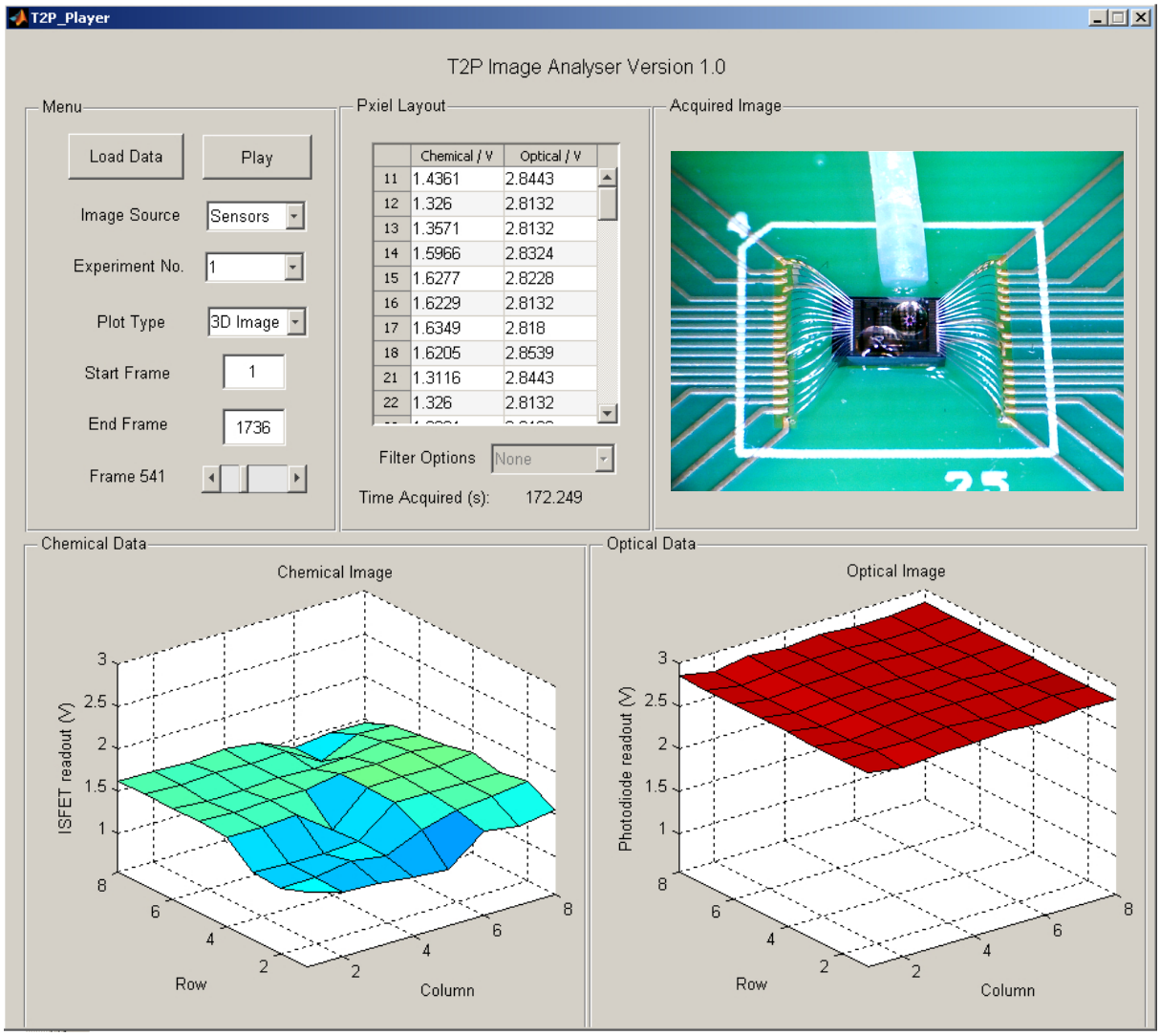


Figure G.1: Lab-on-Chip image and data analyser

Appendix H

Microcontroller Code Listing

H.1 Main Program

```
/** INCLUDES ************************************************************/
#include <18f4680.h> // Modifiable (depends on type of PIC used)
#device high_ints=true
#include <stdio.h> // Required (for I/O functions)
#include <user.h> // Modifiable

/** PIC CONFIGURATION ***************************************************/
#fuses HS,NOWDT,NOPROTECT,NOLVP,NOBROWNOUT,PUT
#use delay(clock=20000000)
#use rs232(BAUD=115200, XMIT=PIN_C7, RCV=PIN_C6)
#use spi(DO=PIN_D3, DI=PIN_B2, CLK=PIN_C4, ENABLE=PIN_C5, LOAD=PIN_D2, ENABLE_ACTIVE=0,
LOAD_ACTIVE=0, MODE=0, BITS=16, BAUD=9600, STREAM=MCP4822)
#use spi(DO=PIN_D5, DI=PIN_B3, CLK=PIN_D6, ENABLE=PIN_D7, LOAD=PIN_D4, ENABLE_ACTIVE=0,
LOAD_ACTIVE=0, MODE=0, BITS=16, BAUD=9600, STREAM=MCP4821)
#use spi(DO=PIN_C2, CLK=PIN_C3, BAUD=9600, LSB_FIRST, MODE=0, stream=LOAD_DATA)
#priority INT_EXT

/** MAIN PROGRAM ********************************************************/
void main(void) {
    while (TRUE) {
        state_machine(); // see user.h, state_machine.c
    }
}
```

H.2 Microcontroller State Machine

H.2.1 Global Variables

```
#define INT0_PER_CYCLE 100 // Number of interrupts per cycle
#define INT1_PER_CYCLE 2 // Number of interrupts per cycle
#define INT2_PER_CYCLE 2 // Number of interrupts per cycle
#define EXTRA_SENS_PULSE 5 // Number of pulses after SYNC_SENS=1

long data_tx[140]; // Output data buffer
char data_rx[140]; // Input data buffer unsigned
int8 px_index; // Pixel count (0 to 63) unsigned
int8 int_count_ind; // Interrupt counter (inductors)
unsigned int8 int_count_sens; // Interrupt counter (sensors)
unsigned int8 sens_count; // Counter for extra pulses
int1 smprdy; // Sample ready (2nd rising CLK_SENS)
int1 eocrdy; // EOC ready
```

H.2.2 Interrupts

EOC Interrupt

```
#INT_EXT FAST
void ext_isr(void) // EOC Interrupt
{
    if(input(PIN_B5)&&(eocrdy==1)) { // If ADC_IN=0
        data_tx[px_index*2]=make16((input_c()&0x3)+0x80,((input_e()<<5)+(input_a()>>1)));
        eocrdy=0; // Set EOC to 0
        ext_int_edge(0,H_TO_L);
        clear_interrupt(INT_EXT);
    }
    if(!input(PIN_B5)&&(eocrdy==1)){ // If ADC_IN=1
        data_tx[px_index*2+1]=make16((input_c()&0x3)+0x80,((input_e()<<5)+(input_a()>>1)));
        eocrdy=0;
        ext_int_edge(0,H_TO_L);
        clear_interrupt(INT_EXT);
    }
    if(eocrdy==0) {
        ext_int_edge(0,L_TO_H);
    eocrdy=1;
    }
}
```

Real Time Data Acquisition Interrupt

```
#INT_TIMER0
void timer0_isr(void) // RT DAQ Interrupt
{
    if(--int_count_ind==0) { // Counts every one period
        output_high(PIN_B4);
        int_count_ind=INT0_PER_CYCLE; // Resets INT counter
    }
    if(int_count_ind==50) {
        output_low(PIN_B4); // Generate clock
    }
    set_timer0(data_rx[3]); // Set clock frequency
    if (!input(PIN_B3)) { // If RTS is set to 1:  User wants to stop
        output_low(PIN_D1);
        disable_interrupts(INT_TIMER0);
        disable_interrupts(GLOBAL);
    }
}
```

Magnetic RAM Initialisation Interrupt

```
#INT_TIMER1 // Magnetic RAM Initialisation Interrupt
void timer1_isr(void)
{
    if (!input(PIN_D0)) { // If SYNC is not 1
        if(--int_count_ind==0) { // Counts every one period
            output_high(PIN_B4);
            int_count_ind=INT1_PER_CYCLE; // Resets INT counter
            }
        if(int_count_ind==10) {
            output_low(PIN_B4); // Generate clock
        }
        set_timer1(65500); //
    }
```

```
    else { // Disable interrupt after initialised
        output_low(PIN_B4);
        output_high(PIN_B2);
        disable_INTERRUPTS(INT_TIMER1);
        disable_INTERRUPTS(GLOBAL);
    }
}
```

Sensor Array Initialisation Interrupt

```
#INT_TIMER2 // Sensor Array Initialisation Interrupt
void timer2_isr(void)
{
    if(--int_count_sens==0) { // Counts every one period
        if (--sens_count==0) {
            smprdy=1; // Currently on smprdy 1
            px_index=0; // Currently on pixel 0
            output_high(PIN_B2); // Set CTS to high
            disable_interrupts(INT_TIMER0);
            disable_interrupts(INT_TIMER2);
            disable_interrupts(INT_EXT);
            disable_interrupts(GLOBAL);
        } else {
            output_high(PIN_D1);
        }
        if (!input(PIN_B1)) { // Reset pulse if SYNC_SENS=0
            sens_count=EXTRA_SENS_PULSE;
        } else {
            px_index=0;
        }
        if (smprdy==1) { // If smprdy=1
            smprdy=0; // Toggle smprdy
            if (!input(PIN_B5)&&!input(PIN_B1)) { // If ADC_IN=0 && SYNC_SENS=0
                px_index++;
            }
        }
        else { // If smprdy=0
            smprdy=1; // Toggle smprdy
        }
        int_count_sens=INT2_PER_CYCLE; // Resets INT counter
    }
    if(int_count_sens==1) {
        clear_interrupt(INT_EXT);
        output_low(PIN_D1); // Generate clock
        if (input(PIN_B1)) {
            px_index=0; // Initialise pixel count at (1,1)
        }
    }
    set_timer2(2);
}
```

H.2.3 State Machine Main Function

```
void state_machine(void)
{
    while (TRUE) {
        if (kbhit()) // If data pushed onto serial buffer {
            output_low(PIN_B2);
            fgets(data_rx); // Get string from PC
```

```
            switch (data_rx[0]) // Switch case menu {
                case 'v' :  // Program voltage references MCP482X
                {
                    ref_voltage();
                    break;
                }
                case 'd' :  // Clock serial data in
                {
                    output_low(PIN_B2); // Force CTS to low
                    load_t2p();
                     break;
                }
                case 'i' :  // Initialise T2P
                {
                    output_low(PIN_B2); // Set CTS to low
                    init_t2p();
                    break;
                }
                case 'f' :  // Continuous magnetic field generator (ind)
                {
                    output_low(PIN_B2);
                    ind_t2p();
                    break;
                }
                case 'a' :  // Real Time Data Acquisition (ind & sens)
                {
                    output_low(PIN_B2);
                    daq_t2p();
                    break;
                }
                default :  // Prints user entered string
                {
                    getc();
                    printf("String entered:  %s\n",data_rx);
                    break;
                }
            } // end switch
        } // end if
    } // end while
} // end state_machine
```

H.2.4 Program DACs

```
void ref_voltage(void)
{
    unsigned long voltage; // 16-bit SPI data word
    switch(data_rx[1]) {
        case 'n' :  // Program ADC_REFN (VOUTA)
            {
                // Shift data_rx[2] left by 8, concatenate data_rx[3]
                voltage=(data_rx[2]*256) + data_rx[3]; // 16-bit SPI data
                spi_xfer(MCP4822,voltage); // Program MCP4822, VOUTA
                break;
            }
        case 'p' :  // Program ADC_REFP (VOUTB)
            {
                // Shift data_rx[2] left by 8, concatenate data_rx[3]
                voltage=(data_rx[2]*256) + data_rx[3];
```

```
                spi_xfer(MCP4822,voltage); // Program MCP4822, VOUTB
                break;
            }
        case 'b' :  // Program ADC_REFBIAS (VREF)
            {
                // Shift data_rx[2] left by 8, concatenate data_rx[3]
                voltage=(data_rx[2]*256) + data_rx[3];
                spi_xfer(MCP4821,voltage); // Program MCP4821, VREF
                break;
            }
    } // end switch
} // end ref_voltage
```

H.2.5 Program Lab-on-Chip

```
void load_t2p(void)
{
    unsigned int8 i=0;
    unsigned int16 ram_data;
    switch(data_rx[1]) {
        case 'g' :  // Load programmable gain amplifiers data
            {
                spi_xfer(LOAD_DATA,data_rx[2],5);
                output_high(PIN_B2); // Indicate PGA ready
                getc();
                output_low(PIN_B2);
                puts(data_rx);
                break;
            } // end case
        case 'i' :  // Load inductor data
            {
                for (i=2;i<130;i=i+2) {
                    if (data_rx[i]>=32 && data_rx[i]<64) { // If 2nd word is 0x00
                        ram_data=make16(data_rx[i],0x00);
                    } else if (data_rx[i]>=64 && data_rx[i]<128) { // If 2nd word is 10
                        ram_data=make16(data_rx[i],10);
                    } else if (data_rx[i]>=128) { // If second word is 13
                        ram_data=make16(data_rx[i],13);
                    } else if (data_rx[i]<32) {
                        ram_data=make16(data_rx[i],data_rx[i+1]); // If no 0x00 or 0xD
                    } spi_xfer(LOAD_DATA,ram_data,10); // Send 10 bits at a time
                } // end for
                output_high(PIN_B2);
                getc();
                output_low(PIN_B2);
                puts(data_rx);
                break;
            } // end case
        case 'p' :  // Load PG-ISFETs data
            {
                for (i=2;i<130;i=i+2) {
                    if (data_rx[i]>=32 && data_rx[i]<64) { // If second word is 0x00
                        ram_data=make16(data_rx[i],0x00);
                    } else if (data_rx[i]>=64 && data_rx[i]<128) { // If 2nd word is 10
                        ram_data=make16(data_rx[i],10);
                    } else if (data_rx[i]>=128) { // If second word is 13
                        ram_data=make16(data_rx[i],13);
                    } else if (data_rx[i]<32) {
```

```
                    ram_data=make16(data_rx[i],data_rx[i+1]); // If no 0x00 or 0xD
                } spi_xfer(LOAD_DATA,ram_data,10); // Send 10 bits at a time
            } // end for
            output_high(PIN_B2);
            getc();
            output_low(PIN_B2);
            puts(data_rx);
            break;
        } // end case
    } // end switch
} // end load_ram
```

H.2.6 Initialise Lab-on-Chip

```
void init_t2p(void)
{
    px_index=0; // Initialise pixel counter
    switch(data_rx[1]) {
        case 'i' :  // Initialise Inductors
            {
                int_count_ind=INT1_PER_CYCLE; // Initialise INT counter
                // Setup Interrupts
                setup_timer_1(T1_INTERNAL|T1_DIV_BY_1);
                enable_interrupts(INT_TIMER1);
                enable_interrupts(GLOBAL);
                set_timer1(65500); //
                getc();
                output_low(PIN_B2);
                output_low(PIN_B4); // CLK_IND idle state 0
                break;
            } // end case
        case 's' :  // Initialise Sensors
            {
                int_count_sens=1; // Initialise INT2 counter
                sens_count=EXTRA_SENS_PULSE; // Initialise 4 CLK_SENS = 1 SYNC_SENS eocrdy=1;
                // Setup Interrupts
                setup_timer_2(T2_DIV_BY_1,0xFA,1); // 10kHz=(20MHz/4)*250*1*2
                enable_interrupts(INT_TIMER2);
                enable_interrupts(INT_EXT);
                enable_interrupts(GLOBAL);
                ext_int_edge(0,L_TO_H); // External interrupt triggers when EOC=0->1
                set_timer2(4); //
                getc();
                puts(data_tx);
                output_low(PIN_D1); // CLK_SENS idle state 0
                output_low(PIN_B2); // Set CTS to low
                break;
            } // end case
    } // end switch
} // end initialise
```

H.2.7 Generate Magnetic Field Pattern

```
void ind_t2p(void)
{
    int_count_ind=INT0_PER_CYCLE; // Initialise INT0 counter
    output_low(PIN_B2);
    indclk_setup();
```

```
    // Setup Interrupts
    while(input(PIN_B3)) { // While MATLAB did not indicate RTS
        enable_interrupts(INT_TIMER0); // CLK_IND pin
        enable_interrupts(GLOBAL);
    }
    puts(data_rx);
}
```

H.2.8 Real Time Data Acquisition with Magnetic Field Pattern Generation

```
void daq_t2p(void)
{
    int_count_ind=INT0_PER_CYCLE; // Initialise INT0 counter
    indclk_setup();
    while(input(PIN_B3)) { // While MATLAB did not indicate RTS
        enable_interrupts(INT_TIMER0); // CLK_IND pin
        enable_interrupts(GLOBAL);
        // Wait for MATLAB to start polling
        if (kbhit()) {
            disable_interrupts(GLOBAL);
            fgets(data_rx); // Get signal from MATLAB
            enable_interrupts(GLOBAL);
            switch (data_rx[0]) // Switch case menu
            {
                case 'i' :
                    {
                        init_t2p();
                        break;
                    } // end case
            } // end switch
        } // end if
    } // end while
    disable_interrupts(INT_TIMER0);
    disable_interrupts(INT_TIMER2);
    disable_interrupts(INT_EXT);
    disable_interrupts(GLOBAL);
}
```

H.2.9 Setup Magnetic RAM Playback Frequency

```
void indclk_setup(void)
{
    switch (data_rx[2]) {
        case 'p' :
            {
                setup_timer_0(RTCC_INTERNAL|RTCC_DIV_1|RTCC_8_BIT);
                set_timer0(data_rx[3]);
                break;
            } // end case
        case 'q' :
            {
                setup_timer_0(RTCC_INTERNAL|RTCC_DIV_2|RTCC_8_BIT);
                set_timer0(data_rx[3]);
                break;
            } // end case
        case 'r' :
            {
                setup_timer_0(RTCC_INTERNAL|RTCC_DIV_4|RTCC_8_BIT);
```

```
                set_timer0(data_rx[3]);
                break;
            } // end case
        case 's' :
            {
                setup_timer_0(RTCC_INTERNAL|RTCC_DIV_8|RTCC_8_BIT);
                set_timer0(data_rx[3]);
                break;
            } // end case
        case 't' :
            {
                setup_timer_0(RTCC_INTERNAL|RTCC_DIV_16|RTCC_8_BIT);
                set_timer0(data_rx[3]);
                break;
            } // end case
        case 'u' :
            {
                setup_timer_0(RTCC_INTERNAL|RTCC_DIV_32|RTCC_8_BIT);
                set_timer0(data_rx[3]);
                break;
            } // end case
        case 'v' :
            {
                setup_timer_0(RTCC_INTERNAL|RTCC_DIV_64|RTCC_8_BIT);
                set_timer0(data_rx[3]);
                break;
            } // end case
        case 'w' :
            {
                setup_timer_0(RTCC_INTERNAL|RTCC_DIV_128|RTCC_8_BIT);
                set_timer0(data_rx[3]);
                break;
            } // end case
        case 'x' :
            {
                setup_timer_0(RTCC_INTERNAL|RTCC_DIV_256|RTCC_8_BIT);
                set_timer0(data_rx[3]);
                break;
            } // end case
    } // end switch
} // end frequency
```

Appendix I

MATLAB Code Listings

I.1 Serial Communication Setup

```
s = serial('COM10'); % Create serial object (PORT Dependent)
set(s,'BaudRate',115200); % Sets Bits per second
set(s,'DataBits',8);
set(s,'Parity','none');
set(s,'StopBits',1);
set(s,'FlowControl','none');
fopen(s); % Open the serial port for r/w
s.RequestToSend='off';
```

I.2 MATLAB State Machine

I.2.1 Program DACs

```
function T2P_DAC(mode,voltage)
% Function progams DAC, and takes in mode describing which reference voltage to
% program, and the user defined biasing level.

if strcmp(mode,'n')==1 || strcmp(mode,'b')==1
    vref=dec2bin(0); % Program ADC- or Reference Electrode
else
    vref=dec2bin(1); $ Program ADC+
end;
if (voltage>2.0475 && voltage<=3.3)
    dn=round(voltage*4096/(2*2.048)); % Compute digital word
    word=[vref,'101',dec2bin(dn,12)]; % Concatenate to give SPI data
else if (voltage<=2.0475 && voltage>=0)
    dn=round(voltage*4096/2.048);
    word=[vref,'111',dec2bin(dn,12)]; % Concatenate to give SPI data
    else
        fclose(s); delete(s); error('Voltage must be between 0 and 3.3V!');
    end;
end;

spi_data1=bin2dec(word(1:8)); % Split 16-bit data word into 2
spi_data2=bin2dec(word(9:16)); % 8-bit words

% Sends serial data to PIC
fwrite(s,['v',lower(mode),uint8(spi_data1),uint8(spi_data2),char(13)]);
```

I.2.2 Program Lab-on-Chip

Program Magnetic Field Pattern

```
function [ram_data]=T2P_Magnetic(px_pol,px_mag,row,col,ram_data,ram_num)
% Function to program each RAM element in magnetic controller.  Takes in
% pixel polarity, magnitude, row and column.  Updates ram_num element
% in 64x4 cell array ram_data containing magnetic field pattern data.

if px_pol == -1 % Converts polarity into binary bit
    bin_pol = dec2bin(1);
else
    bin_pol = dec2bin(0);
end;
px_flux=px_pol*px_mag; % Calculate flux
% Compute Binary word
word=[dec2bin(row-1,3) dec2bin(col-1,3) bin_pol dec2bin(px_mag,3)];
ram_data(ram_num,:)={row,col,px_flux,word};
```

Pre-process RAM Data

```
function [post_data]=T2P_RAM(pre_data)
% Function to align ram bits.  Takes in a 64x1 data vector.  Processes data
% by ordering and flipping bits, returning a 64x1 processed vector.

if iscell(pre_data) % For magnetic field pattern data
    Dn=fliplr(bin2dec(ram_data(:,4))'); % Reverse order of bits
    for i=1:4 % Reverse order for odd rows
        Dn(2*(i-1)*8+1:((2*(i-1))+1)*8)=fliplr(Dn(2*(i-1)*8+1:((2*(i-1))+1)*8));
    end;
else % Calibration data is already flipped in calibration function
    Dn=round(pre_data*1024/3.3); % For calibration data
end;
Dn_bin=dec2bin(Dn,10); % Compute 10 bit word

% Flip binary bits because RAM snakes around
for i=1:2:7
    index((i+1)/2,:)=(i-1)*8+1:i*8;
    Dn_bin(index((i+1)/2,:),:)=fliplr(Dn_bin(index((i+1)/2,:),:));
end;
% Format bit stream to PIC
Dn1=(bin2dec(Dn_bin(:,1:2))+16);
Dn2=(bin2dec(Dn_bin(:,3:10)));

% Correct errorneous bits
err0=find(Dn2==0); % Finds bytes containing all eight 0s
err10=find(Dn2==10); % Finds bytes containing terminator 10
err13=find(Dn2==13); % Finds bytes containing terminator 13
Dn1(err0)=Dn1(err0)+32; % Indicate Dn2 contains all 0 (add 2^5)
Dn1(err10)=Dn1(err10)+64; % Indicate Dn2 contains 10 (add 2^6)
Dn1(err13)=Dn1(err13)+128; % Indicate Dn2 contains 13 (add 2^7)
Dn2(err10)=63;
Dn2(err0)=63; % Replace errorneous bits with 0000111111
Dn2(err13)=63;

for i=1:64 % Store values in array
    post_data(i*2-1)=Dn1(i); % First 5 bits
    post_data(i*2)=Dn2(i); % Next 5 bits
end;
```

Load Data

```
function T2P_LoadRam(Gisfet,Gopto,isfet_data,ind_data)
% Function loads lab-on-chip RAM with PGA gains Gisfet and Gopto, ISFET data,
% magnetic field pattern ind_data, through global serial object s.

% Process PGA data and load PGA gains
Gisfet_vec=[10 6 5 1]; % Initialise possible gains
Gisfet_bin=dec2bin(find(Gisfet_vec==Gisfet)-1,2); % Compute digital code
Gopto_bin=dec2bin(Gopto-1,3); % Compute digital code
Gpga=bin2dec([Gopto_bin Gisfet_bin])+64;

fwrite(s,['dg',uint8(Gpga),char(13)]); % Program PGA
while(strcmp(s.PinStatus.ClearToSend,'on')==0) % Wait for PIC signal
end;
fprintf(s,'A'); % Reply to microcontroller

fwrite(s,['di',uint8(ind_data),char(13)]); % Program B-field
while(strcmp(s.PinStatus.ClearToSend,'on')==0) % Wait for PIC signal
end;
fprintf(s,'A'); % Reply to microcontroller

fwrite(s,['dp',uint8(isfet_data),char(13)]); % Program PG-ISFETs
while(strcmp(s.PinStatus.ClearToSend,'on')==0) % Wait for PIC signal
end;
fprintf(s,'A'); % Reply to microcontroller
```

I.2.3 Read Sensor Data

```
function [w]=T2P_Read(s)
% Function to read data from microcontroller, through serial object s.

w_chem=1.65*ones(1,64); % Initialise chemical data
w_opto=1.65*ones(1,64); % Initialise optical data
k=1; % Initialise counter
w_chem=zeros(8,8); % Initialise chemical image
w_opto=zeros(8,8); % Initialise optical image
w(1)=fread(s,1,'uint16');
if (w(1)~=2573 && s.BytesAvailable>0) % If data is valid
    while(w(k)~=2573 && k<128 && s.BytesAvailable>0)
        k=k+1;
        w(k)=fread(s,1,'uint16'); % Read data from the microcontroller
    end;
    if (k==128 && w(end)~=2573) % If we have 128 elements (64*2 pixels) and w(end)
        discard=fgets(s); % Discard all extra data pushed onto PC buffer
    end;
end;
```

I.2.4 Display Sensor Data

```
function [w_chem,w_opto]=T2P_Plot(w,refp,refn)
% Function to display sensor data.  Takes in sensor data (w) read from
% microcontroller, ADC+ (refp) and ADC- (refn).  Returns sensor images.

% Initialise chemical/optical data and images
w_chem=zeros(1,64); w_opto=zeros(1,64);
[y,x]=meshgrid(1:1:8);

% Extract and process binary data from microcontroller
```

```
w_bin=dec2bin(w(1:end),16); % Convert to decimal
w_dec=bin2dec(w_bin(:,7:16))*((refp-refn)/1023)+refn; % Scale data wrt ADC
w_chem=w_dec(1:2:end); % Extract chemical data
w_opto=w_dec(2:2:end); % Extract optical data

for i=1:8 % Order chemical data into array format
    w2_chem(i,1:8)=w_chem((i-1)*8+1:i*8);
end;
subplot(1,2,1); % Plot chemical image
surf(x,y,w2_chem,'CDataMapping','scaled'); rotate3d on;
xlabel('Column'); ylabel('Row'); zlabel('Chemical readout / V');
axis([1 8 1 8 refn refp]); caxis([refn,refp]);
title('Chemical Image');

for i=1:8 % Order optical data into array format
    w2_opto(i,1:8)=w_opto((i-1)*8+1:i*8);
end;
subplot(1,2,2); % Plot optical image
surf(x,y,w2_opto,'CDataMapping','scaled'); rotate3d on;
xlabel('Column'); ylabel('Row'); zlabel('Optical readout / V');
axis([1 8 1 8 refn refp]); caxis([1.65,3.3]);
title('Optical Image');
```

I.2.5 Initialise Lab-on-Chip

```
function T2P_Initialise(clk)
% Function which initialises lab-on-chip RAM pointers and sensory array.
% Takes in clk which determines which sub-system to initialise.
% Sensor initialisation will involve polling the sensor array,
% acquiring and displaying data from the lab-on-chip.

fwrite(s,['i',lower(clk),char(13)]); % State machine command
if (strcmp(clk,'s')==1) % If initialising sensors
    while(strcmp(s.PinStatus.ClearToSend,'on')==0) % Wait for PIC signal
    end;
    fprintf(s,'Z'); % Reply to microcontroller
    [w]=T2P_read(s); % Get data from serial buffer
    [h,w_chem,w_opto]=T2P_Plot(w,pg_volt,refp,refn);
else
    while(strcmp(s.PinStatus.ClearToSend,'on')==0)) % Wait for PIC signal
    end;
    fprintf(s,'Z'); % Reply to microcontroller
end;
```

I.2.6 Sensor Calibration

```
function T2P_Calibrate(Gisfet,Gdiode,ind_data,refp,refn)
% Function to calibrate PG-ISFET array.  Takes in PGA gains and magnetic field
% pattern data to be loaded, and ADC+ and ADC- for displaying sensor readout
% images with each iteration.

% Input constant PG voltage to all ISFETs
pg_volt=1.65*ones(1,64); % Use x0 = constant PG voltages
[isfet_data]=T2P_PGram(pg_volt); % Orientate wrt RAM
T2P_LoadRam(Gisfet,Gdiode,isfet_data,ind_data); % Load T2P RAM

% Poll Sensors once
fwrite(s,['is',char(13)]);
```

```
while(strcmp(s.PinStatus.ClearToSend,'on')==0) % Wait for PIC signal
end;
[w]=T2P_Read(s); % Read sensor readout data from microcontroller
[w_chem,w_opto]=T2P_Plot(w,refp,refn)

% Compute error
d{p}=fliplr(w_chem'-1.65);
for i=1:4 % Orientate wrt RAM buffer
    d{p}(2*(i-1)*8+1:((2*(i-1))+1)*8)=fliplr(d{p}(2*(i-1)*8+1:((2*(i-1))+1)*8));
end;

while (1/64)*(d{p}*d{p}')>Gisfet*1e-5 % Run algorithm until mse < Gisfet*1e-5
    % Computer new PG voltages
    pg_volt=pg_volt+d{p}/Gisfet; % Research direction d scaled by step 1/Gisfet
    [isfet_data]=T2P_PGram((pg_volt)); % Load T2P RAM
    T2P_LoadRam(Gisfet,Gdiode,isfet_data,ind_data);

    % Poll Sensors once
    fwrite(s,['is',char(13)]);
    while(strcmp(s.PinStatus.ClearToSend,'on')==0) % Wait for PIC signal
    end;
    [w]=T2P_Read(s); % Read sensor readout data from microcontroller
    [w_chem,w_opto]=T2P_Plot(w,refp,refn);

    % Compute new error
    p=p+1; % Increment iteration counter
    d{p}=fliplr(w_chem'-1.65);
    for i=1:4 % Orientate wrt RAM buffer
        d{p}(2*(i-1)*8+1:((2*(i-1))+1)*8)=fliplr(d{p}(2*(i-1)*8+1:((2*(i-1))+1)*8));
    end;
end;
```

I.2.7 Real Time Data Acquisition

```
function T2P_DAQ(s,vid)
% Function to perform real time data acquisition from sensor array.  Acts on
% global variable sens_state which controls acquisition start and stop,
% communicates through serial object s and acquire microscope images from vid

if sens_state==1
    start(vid); % Start image acquisition from microscope
    while(sens_state==1)
        imaq_flag=1; % Acquire and plot data
        fwrite(s,['is',char(13)]); % Write state machine commmand to PIC
        while(strcmp(s.PinStatus.ClearToSend,'on')==0) % Wait for PIC signal
            if imaq_flag==1
                imaq_img=getsnapshot(vid); % Take 1 snapshot from microscope
                image(imaq_img); % Display microscope image
                imaq_flag=2; % toggle imaq flag
            end; % end if
        end; % end while
        [w]=T2P_Read(s); % Read sensor readout data from microcontroller
        [w_chem,w_opto]=T2P_Plot(w,refp,refn); % display data images
        if sens_state==0
            stop(vid);
        end; % end if
    end; % end while
end; % end if
```

I.2.8 Magnetic Field Pattern Generation

```
function T2P_ind(s,vid)
% Function to generate magnetic field pattern.  Acts on global variable
% imaq_state which controls field pattern generation start and stop.  Takes in
% serial object s and video object vid to image sensor array.

if imaq_state==1;
    s.RequestToSend='on'; % Set RTS to 1.  PIC generates clock while RTS=1
    fwrite(s,['fz',uint8(a),char(13)]);
    start(vid);
    imaq_img=getsnapshot(vid);
    while imaq_state==1
        imaq_img=peekdata(vid,1); % Take microscope snapshot of chip
        image(imaq_img); % Display image capture
        if imaq_state==0
            stop(vid);
            s.RequestToSend='off'; % Set RTS to 0
        end; % end if
    end; % end while
end; % end if
```

I.2.9 Real Time Data Acquisition with Magnetic Field Pattern Generation

```
function T2P_DAQInductors(s,vid)
% Function to acquire data from sensor array and generate magnetic field pattern
% simultaneously.  Acts on global variable rtdaq_state which controls acquisition
% and generation start and stop.  Takes in serial object s and video object vid.

if rtdaq_state==1
    s.RequestToSend='on'; % Set RTS to 1.  PIC stops when RTS is 0
    start(vid);
    fwrite(s,['ax',uint8(a),char(13)]); % write state machine command to PIC
    while(rtdaq_state==1)
        imaq_flag=1; % Acquire and plot data
        fwrite(s,['is',uint8(a),char(13)]); % Write state machine commmand to PIC
        while(strcmp(s.PinStatus.ClearToSend,'on')==0) % Wait for PIC signal
            if imaq_flag==1
                imaq_img=getsnapshot(vid); % Take 1 microscope snapshot of array
                image(imaq_img); % Display snapshot
                imaq_flag=2;
            end; % end if
        end; % end while
        [w]=T2P_Read(s); % Read sensor readout data from microcontroller
        [w_chem,w_opto]=T2P_Plot(w,refp,refn); % display data images
        if rtdaq_state==0
            s.RequestToSend='off'; % Set RTS to 0, stop process.
            stop(vid);
        end; % end if
    end; % end while
end; % end if
```