PNNL-17213

Pacific Northwest National Laboratory
Operated by Battelle for the U.S. Department of Energy

DCE Bio Detection System Final Report

MA Lind
CR Batishko
GP Morgen
SL Owsley Jr
GC Dunham
MG Warner
JA Willett

December 2007

Prepared for the U.S. Department of Energy
under Contract DE-AC05-76RL01830

PACIFIC NORTHWEST NATIONAL LABORATORY
operated by
BATTELLE
for the
UNITED STATES DEPARTMENT OF ENERGY
under Contract DE-AC05-76RL01830

Printed in the United States of America

**Available to DOE and DOE contractors from the
Office of Scientific and Technical Information,
P.O. Box 62, Oak Ridge, TN 37831-0062;
ph: (865) 576-8401
fax: (865) 576-5728
email: reports@adonis.osti.gov**

**Available to the public from the National Technical Information Service,
U.S. Department of Commerce, 5285 Port Royal Rd., Springfield, VA 22161
ph: (800) 553-6847
fax: (703) 605-6900
email: orders@ntis.fedworld.gov
online ordering: http://www.ntis.gov/ordering.htm**

This document was printed on recycled paper.
(9/2003)

PNNL - 17213

DCE Bio Detection System Final Report

Michael Lind
Charles Batishko
Gerry Morgen
Stan Owsley
Glen Dunham
Marvin Warner
Jesse Willett

December 31, 2007

Prepared for the USAF/Brooks City Base
under MIPR AFMC277 No.RAF27704670060
with the U.S. Department of Energy
under contract DE-AC05-76RL01830

Pacific Northwest National Laboratory
Richland, WA 99352

Table of Contents

List of Figures

Introduction

The DNA Capture Element (DCE) Bio-Detection System (Biohound) was conceived, designed, built and tested by PNNL under a MIPR for the US Air Force under the technical direction of Dr. Johnathan Kiel [1] and his team at Brooks City Base in San Antonio Texas. The project was directed toward building a measurement device to take advantage of a unique aptamer based assay developed by the Air Force for detecting biological agents. The assay uses narrow band quantum dots fluorophores, high efficiency fluorescence quenchers, magnetic micro-beads beads and selected aptamers to perform high specificity, high sensitivity detection of targeted biological materials in minutes. This final report summarizes and documents the final configuration of the system delivered to the Air Force in December 2008.

While the details of the assay will not be discussed here, the basic detection process is simple in concept. Detection aptamers (short single strand DNA segments) are formulated for specific target molecules and attached to functionalized quantum dot fluorophores along with an appropriate spaced non-radiative quencher. In the absence of a complimentary DNA fragment supplied by the target biological, the proximity of the quencher and the quantum dot prevent the quantum dot from fluorescing. When a complimentary DNA fragment attaches itself to the aptamer with the fluorophore, the physical separation of the quencher and the quantum dot changes sufficiently to allow fluorescence in the presence of a suitable excitation source. Thus with no target molecule present, the solution remains dark. If a target is present, the solution glows when illuminated by the proper light source. If there is sufficient target present one can see the glow visually. If only trace quantities are present it requires a sensitive detector and a concentrator to see the fluorescence. The DCE Bio-Detection System described in this report fills that trace detection function.

The emitted light indicates the presence of the target molecule and the number of photons detected is proportional to the concentration of the target, the number of binding sites on the target, the efficiency of binding, the efficiency of de-quenching, the temperature, the optical obscuration fraction and a host of other factors. So while the biological sensitivity is target specific, the instrument sensitivity is more easily calibrated as a function of the minimum number of detectable quantum dots. For this DCE instrument the nominal detection noise threshold is estimated at 5 -10 femto moles of quantum dots. The correlation to the biological target threshold has not yet been determined.

[1] AFMC277 No.RAF27704670060 dated 18 April 2004

A key to high sensitivity detection is concentration of the target. In this DCE instrument, concentration is achieved using magnetic micro beads with affinity binders attached. There are many variations of this scheme possible. In one simple variation the beads act as collectors for classes of organic molecules. When put into a buffer solution and sprayed on an area to be analyzed, they will preferentially attract the biological materials of interest essentially magnetizing those materials. When the sprayed area is sampled using any of a number of standard sample collection methods, mixed with the aptamer/quantum dot solution, and subsequently flowed past a strong magnet, the magnetic beads are attracted to and localized by that magnet for detection. This magnetic concentration occurs in the DCE cartridge where volumetric concentration approaching 1000 can be achieved.

The DCE Detection System incorporates both the optics/electronics and the collection cartridge into a single measurement system. Figure 1 is a photograph of the complete measurement system showing the detection unit and a vacuum sealed package containing four measurement cartridges.

Figure 1. DCE Bio-Detection System showing the measurement module and a package of vacuum sealed cartridges.

The DCE Bio-Detection System while shown in the preferred mode in Figure 1 supports several other measurement modes. One alternative uses a glass plate that has a 5 μl well to hold a 5 μl sample that is covered with a microscope slip cover. Another alternative uses a second specially designed flip top lid that houses a removable rectangular cuvette centered over the detector port. Using this method, solutions can be put into a cuvette for incubation, placed over a magnet to concentrate any magnetic particles at the bottom of the cuvette. The cuvette is then placed into the special lid which places the bottom of the cuvette over the measurement area and a light tight cover is placed over the cuvette. This allows the measurement of the contents at the bottom of the cuvette only. The preferred method uses a flow cell cartridge that draws in a solution from a syringe via an injection septum. The cartridges have a small integral magnet attached over the measurement area that attracts magnetic particles to concentrate them over the measurement port. The cartridge has the advantage of being a sealed system that can be externally decontaminated, retained for subsequent forensic analysis, and is inherently chain of custody friendly.

There are many possible extensions to this prototype measurement system. In principle the system can be used for any type of sandwich assay employing an affinity binder a fluorophore reporter (dye or quantum dot) and a magnetic particle. For example it has been used successfully on a botulinum toxin assay using antibodies, dyes and magnetic nano-particles. The device could also be extended to multiple simultaneous assays with changes to the optical package and multiple color reporters.

Operating Procedure

The DCE Bio-Detection System is designed to be easy to operate with limited training and minimal manual dexterity. The measurement cartridges may be loaded by syringe injection through its vacuum septum outside of the measurement unit or inside the measurement unit. The cartridge is inserted into the measurement unit by lifting the hinged cover and placing it with the magnet side up in the fitted recess. Closing the cover provides optical background isolation for maximum sensitivity. Figure 2 shows a cartridge mounted in the proper configuration.

Figure 2. Cartridge in measurement unit prior to lid closure.

There are four momentary pushbutton switches that control the actual measurement sequence., one for turning on and off the power, and three function or mode switches. There is also a liquid crystal alphanumeric display for indicating system status and four LEDs which indicate the relative level of target agent detected. These switches and displays are shown in the Figure 3 below.

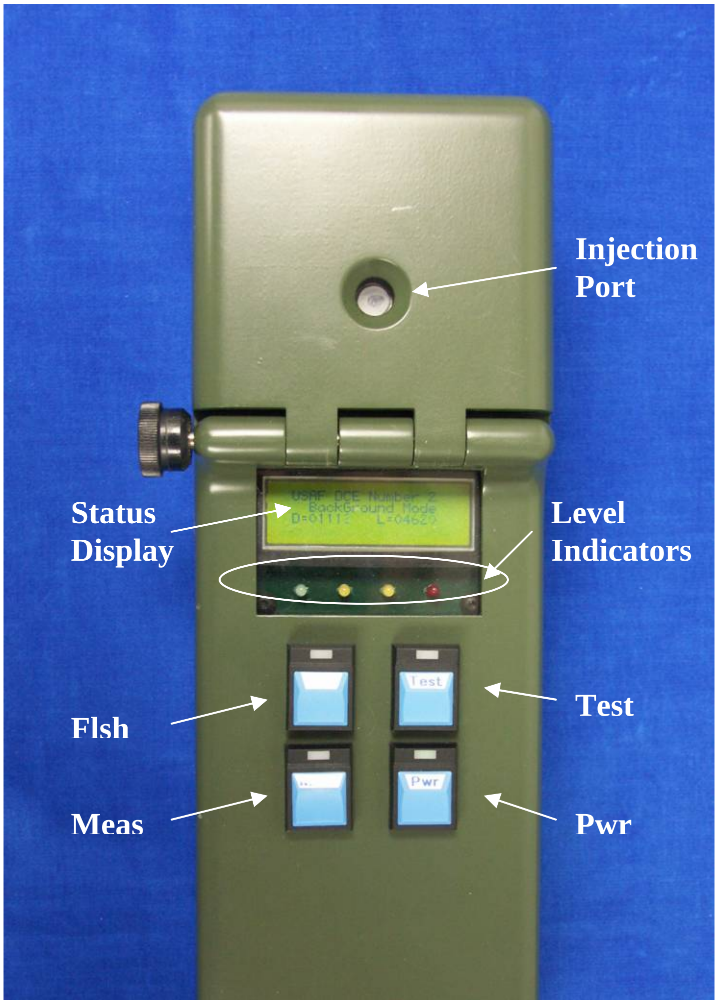

Figure 3. Top view of the DCE measurement apparatus showing the injection port, the LCD status display, the four LED level indicators and the four control buttons on the top side of the measurement instrument.

The switches are labeled:

Pwr – Power On / Off Switch (lower right)
Meas – Measure Mode (lower left)
Test – Test Mode (upper right)
Flsh – Flash Light Mode (upper left)

After applying power, either using batteries or the laboratory power supply, the DCE is turned on by pressing the lower right ***Pwr*** switch for 1 to 2 seconds. The red LED in the lower left ***Meas*** switch will blink on then off during the initialization process which indicates the instrument is functioning properly and has entered the ***startup mode***. Once the DCE is initialized and the power latched, the LCD displays the startup banner.

After a 5 second delay, the ***background mode*** displays raw data values. The data displayed is for the direct detector signal, D, the loc-in amplifier signal, L. The lock-in amplifier signal is the rectified AC component of the fluorescence signal that is synchronously modulated with the illumination source, rejecting much of the asynchronous background noise. Although L is the primary signal of interest, D provides a diagnostic indication of electronics malfunction, DC drift, stray light problems and detector saturation problems Normal values in the ***background mode*** should look something like D = 00548 L = 02370 . These numbers derived from a 16-bit analog to digital converter (ADC) which has a range 0 - 65535 (0 to 2.048 volts.

The ***background mode*** displays during the warm up period. Depending on environmental conditions, up to 20 minutes warm-up is required to obtain low drift readings after the device is first powered up. It is advisable to place an unused cartridge in the measurement apparatus to obtain the most representative background (null sample) readings.

Pressing and releasing the lower left ***Meas*** switch again transitions from the ***background mode*** to the ***measure mode*** saving the background readings to be used as an offsets. The background readings are subtracted from the new D and L values being acquired with the filled cartridge. Simultaneously the four indicator light emitting diodes (LEDs) [Green, Yellow, Yellow, and Red] will activate. These LEDs are used to indicate a range of D and L values. The ranges are listed below and were selected based upon quantum dot calibration data.

Green:	L = 0 to 199
Yellow:	L = 200 to 599
Both Yellow:	L = 600 to 1999
Red:	L > 2000

Pressing and releasing the lower left ***Meas*** switch again transitions back to the ***background mode*** to prepare for another measurement cycle.

Pressing and releasing the upper right ***Test*** switch transitions from the ***background mode*** to the ***test mode***. This mode is used for general testing and setup, such as checking flow cells or cuvettes for fluorescence levels before being used in a measurement procedure.

Remember that finger oils, cigarette smoke, and a host of other common contaminates are highly fluorescent and can significantly interfere with these sensitive measurements.

The ***test mode*** is similar to the ***measure mode***, except the LEDs are not active. When going from ***background mode*** to ***test mode***, the D and L values at the time the switch is pushed are used as an offset, and subtracted from the new D and L values being acquired. The values displayed can go negative if there is fluorescent material in the optics field of view when the ***Test*** switch is pressed and then removed during the ***Test*** mode. Pressing and releasing the ***Test*** switch again reverts back to the ***background mode***.

Note that the ***Meas*** and ***Test*** switches transition to the ***measure mode*** and ***test mode*** only from the ***background mode***. If the ***Test*** switch is pressed when in the ***measure mode***, DCE will go to the ***background mode*** and ***Test*** must be pressed again to transition to the ***test mode***. The same operation holds for the ***Meas*** switch if in the ***test mode***.

Pressing and releasing the upper left ***Flsh*** switch transitions to the ***flashlight mode***. This turns off the UV LED used for ***measure mode*** and ***test mode*** and turns on a UV LED mounted in the end of the DCE box – the same end as the DC power plug. The ***Meas*** and ***Test*** switches do not function in the ***flash light mode***. This mode could be used to search for areas or objects that fluoresce. Pressing the ***Flsh*** switch again transitions out of ***flash light mode*** and reverts to ***background mode***.

Power Options

The unit can be powered by batteries for remote operation or by an external laboratory DC power supply for doing extended time frame measurements in the laboratory. The laboratory power supply provided is model Instek GPS-4303.

Battery operation uses three 9 volt batteries. Lithium batteries are preferred for their extended capacity and wider temperature capabilities, but standard alkaline cells will work. To install the batteries, remove the small cover at the lower end of the rear of the case and insert them into the holders.

Batteries can remain in the unit when using the external laboratory power supply. The instrument detects when the external power source is being used and draws from the external source. When the battery only is present, the instrument will automatically turn off after 15 minutes if inactivity (i.e. no switches have been pushed). If the external power supply is being used and the batteries are installed, the instrument will automatically switch to battery operation if the external power source is removed without turning the unit off. This allows for extended warm up periods prior to mobile use.

The laboratory supply is used to generate +10.0 volts & -10.0 volts for the electronics, and a separate 10.0 volts for the UV LED operation. After turning on the power supply, check and set the following voltages:

CH1 = 10.0 volts
CH2 = 10.0 volts
CH4 = 10.0 volts [upper left slide switch toggles CH2 - CH4 display]

A power cable with connectors, red/black banana plugs for connections to the lab power supply and 5 pin connector for the DCE box, is provided. The Blue-Black wire pair connects to CH1, with the red plug to the red terminal and the black plug to the black terminal. The White-Black wire pair connects to CH2, with the red plug to the red terminal and the black plug to the black terminal. The Red-Black wire pair connects to CH4, with the red plug to the red terminal and the black plug to the black terminal. Internal wiring determines which supply is positive and which is negative.

After turning on the Instek GPS-4303 POWER Switch, the OUTPUT button must be pushed to activate the power supply output terminals. When using the lab power supply, turn it on, check that the 3 output voltages are all 10.0 volts, turn on the output, then plug the DC power connector into the DCE. Turn on the DCE by pressing the DCE power switch – ***Pwr***. Turn off the DCE by again pressing the DCE power switch – ***Pwr***.

Optical Design

The DCE is specifically designed to detect and measure the fluorescence of the Evident Technologies Fort Orange Qdots in a solution at a wavelength 600nm. All components were selected to maximize performance without regard to individual component cost. Maximum throughput and minimum out of band transmission, maximum stray light rejection, minimum component fluorescence, minimum polarization sensitivity and minimum internal scatter were the goals driving the design and selection of components. The simple optical system layout is shown in block form in Figure 4.

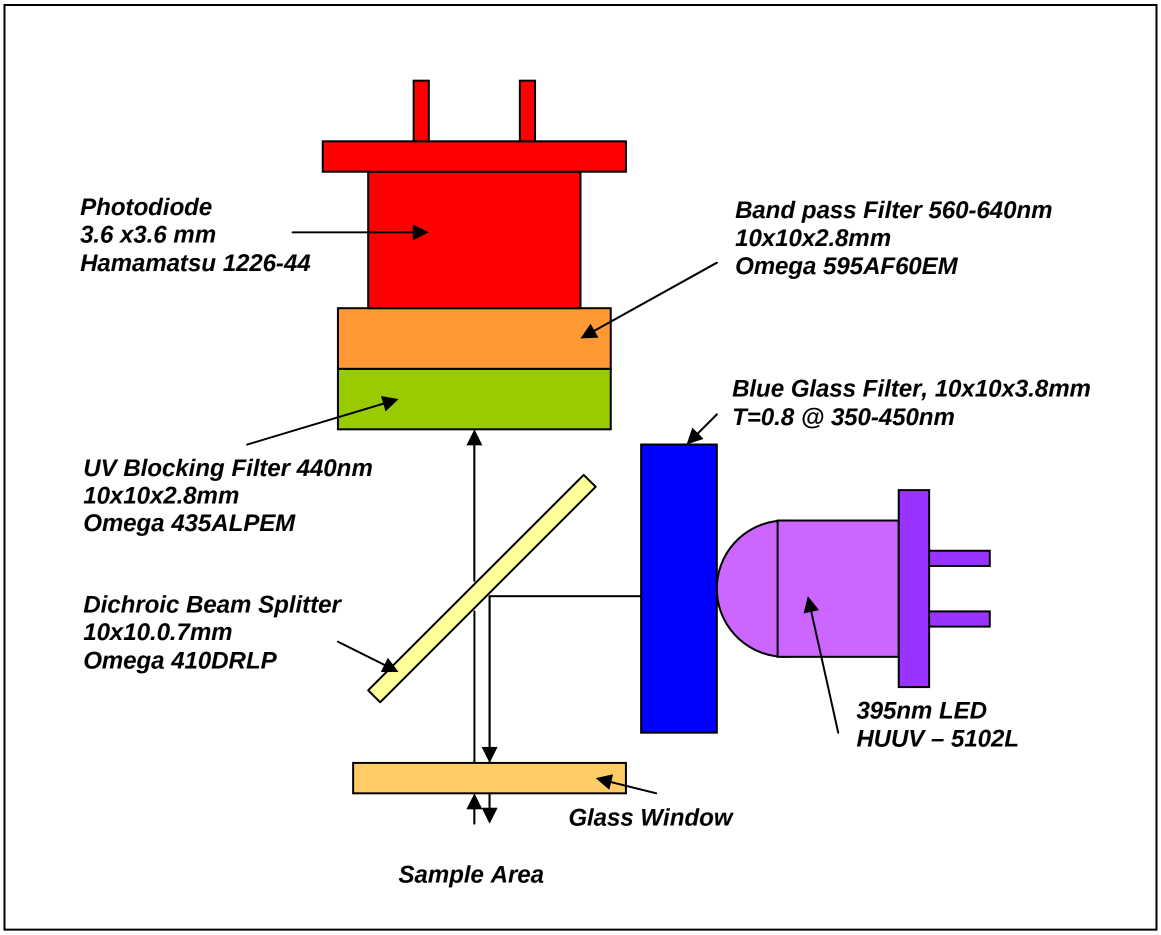

Figure 4. Optical block diagram for DCE Bio-Detection System.

The optical source to stimulate fluorescence is a UV LED that operates at 395 nm. It is operated in pulsed mode running at 1 kHz with a duty cycle of 12% in order to take advantage of the stray light rejection of lock-in detection. The output of the diode is filtered to remove long wavelength components from the source which will compromise the low level performance of the system. The 395nm UV light which maximizes the fluorescence excitation of the dyes and quantum dots used in the assays is directed onto the sample area using a 45 degree dichroic beam splitter. The resulting diffuse downshifted fluorescence passes through the beam splitter and a series of blocking/band pass filters that are centered on the Fort Orange response of 600 nm. The low pass filter absorbs reflected UV light above 430 nm that might be scattered from the cartridge or the carrier fluid. An adjacent band pass filter narrows the pass band to the maximum fluorescence region (580nm) in order to further reduce ambient light effects. A solid state Hamamatsu photodiode is the sensor selected for high sensitivity and a large active area.

Mechanical Design

The mechanical design for the instrument is driven by the following considerations:

- The optics assembly is protected from stay light to the greatest extent possible
- An easily accessible battery compartment
- Simple operational controls for use in conditions of impaired dexterity
- Indexed sample cartridge holder easily assessable with impaired dexterity
- Visible yet clandestine displays
- Sealed for rapid decontamination via bleach solution emersion
- Light weight, single hand carry and operation

Figure 5 below is a dimensioned drawing of the DCE instrument with top and side views. The bottom cover is not shown, but fits in two pieces to cover the electronics and the battery compartment. The electronics/optics compartment is o-ring sealed.

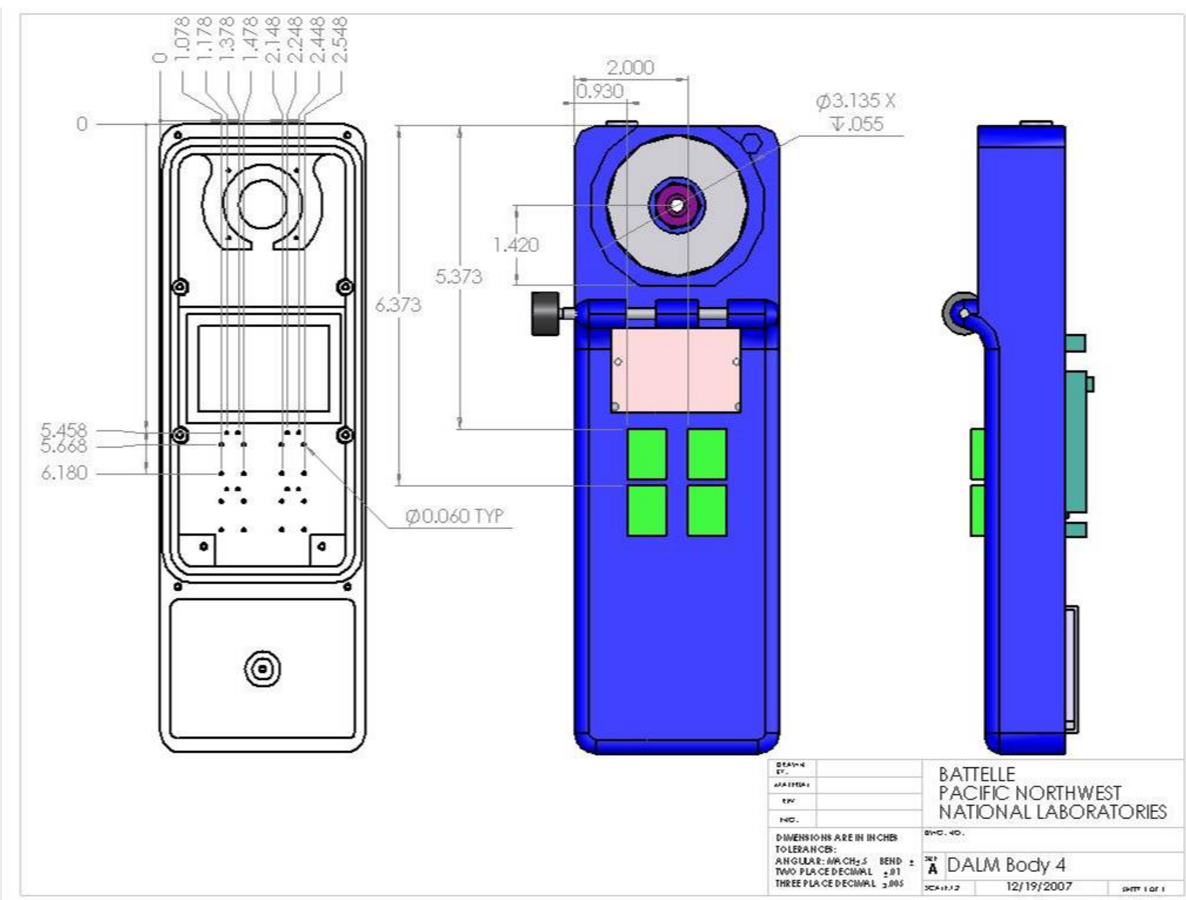

Figure 5. Dimensioned drawing of the DCE instrument case.

The bottom side of the instrument showing the battery compartment is shown in Figure 6.

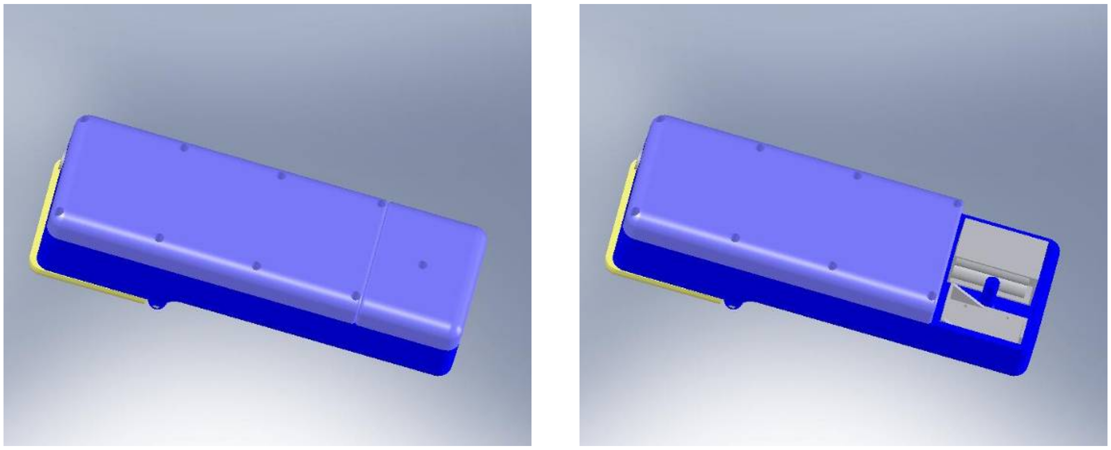

Figure 6. Bottom view of DCE instrument showing battery compartment.

Two flip top sample covers are supplied with the instrument. One is for covering the cartridge, the second is to hold a cuvette. The dimensional drawings of cartridge cover showing the fill port access is shown in Figure 7 and cuvette holder in Figure 8.

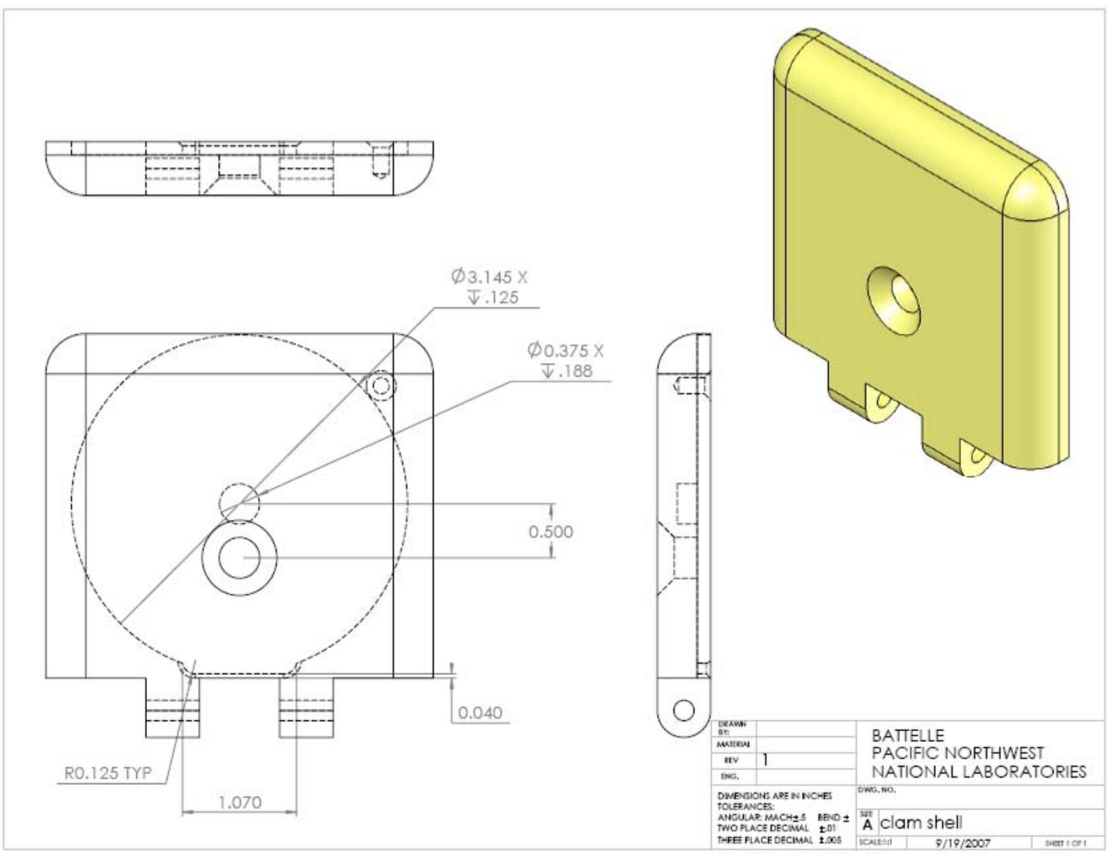

Figure 7. Dimensional drawing of DCE flip top cartridge cover

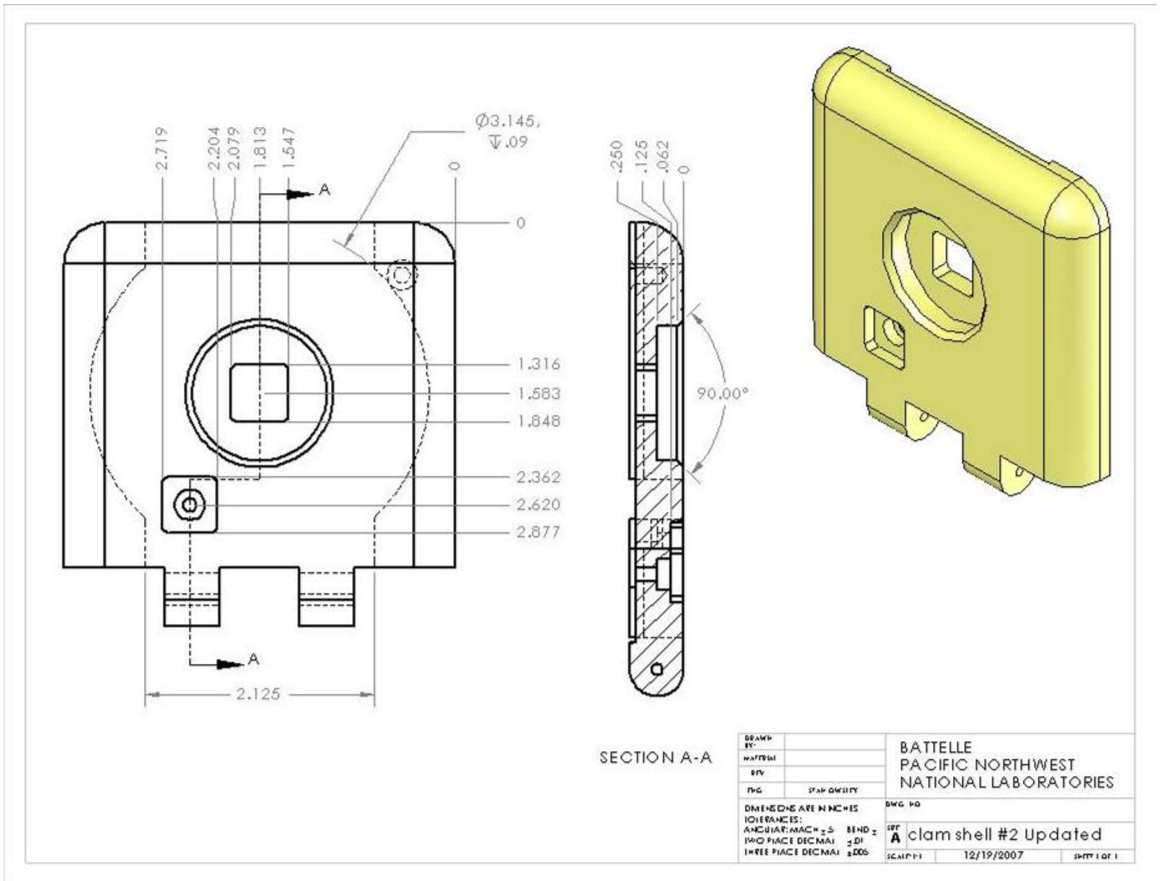

Figure 8. Dimensional drawing of DCE cuvette holder.

Figure 9 shows the cuvette holder installed on the DCE instrument. The cylinder is an accompanying light shield place over the cuvette after insertion.

Figure 9. Installed cuvette holder with adjacent light shield.

Electronics Design

The schematics for the electronics supporting the DCE instrument are shown in Figure 10-14. They represent the most recent *red lined* drawings for the delivered prototypes. All components are commercial off the shelf devices at the time the prototypes were assembled. Full sized drawing in *pdf* files are available on the CD that accompanies this report.

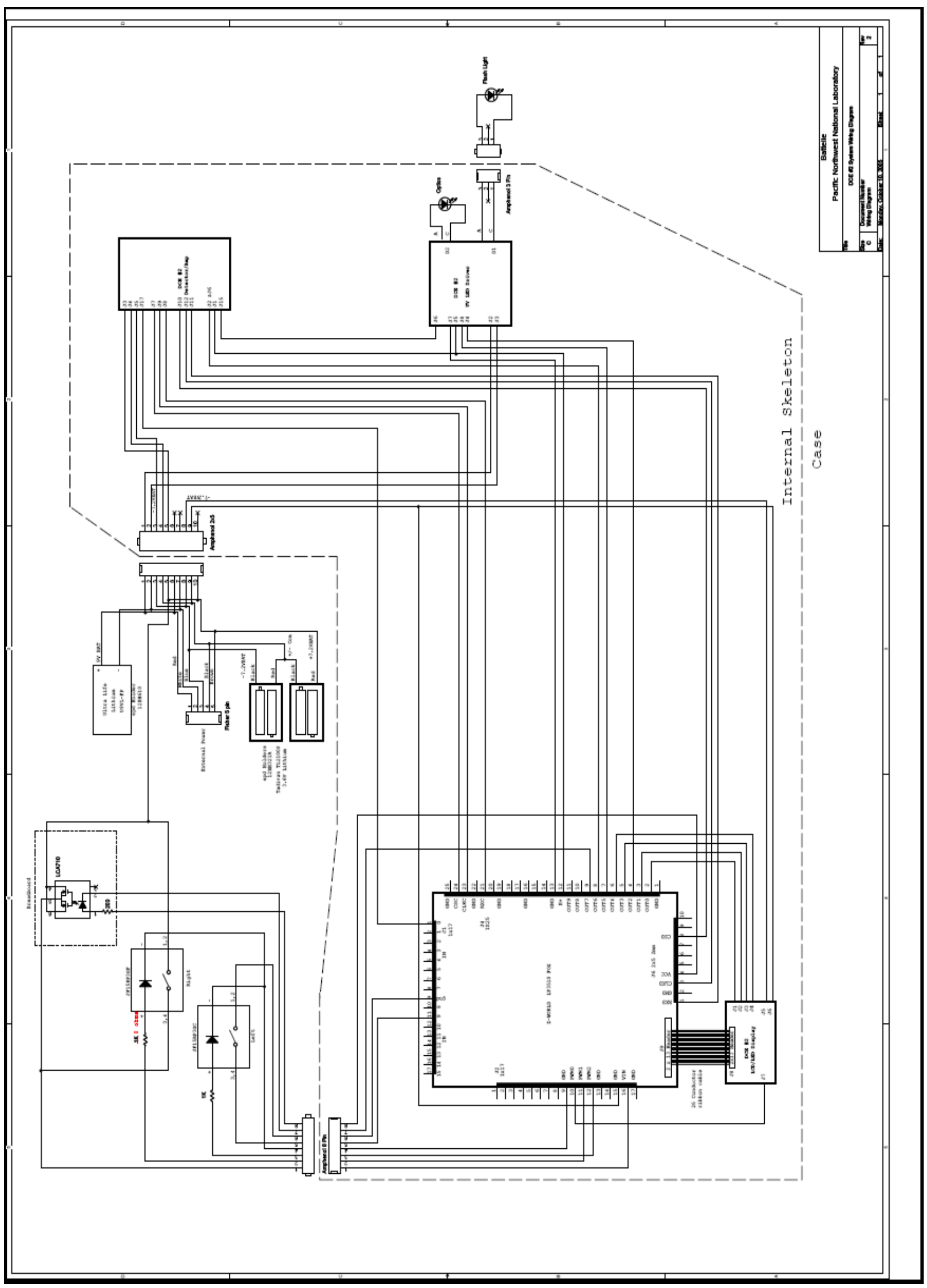

Figure 10. DCE electronics wiring diagram.

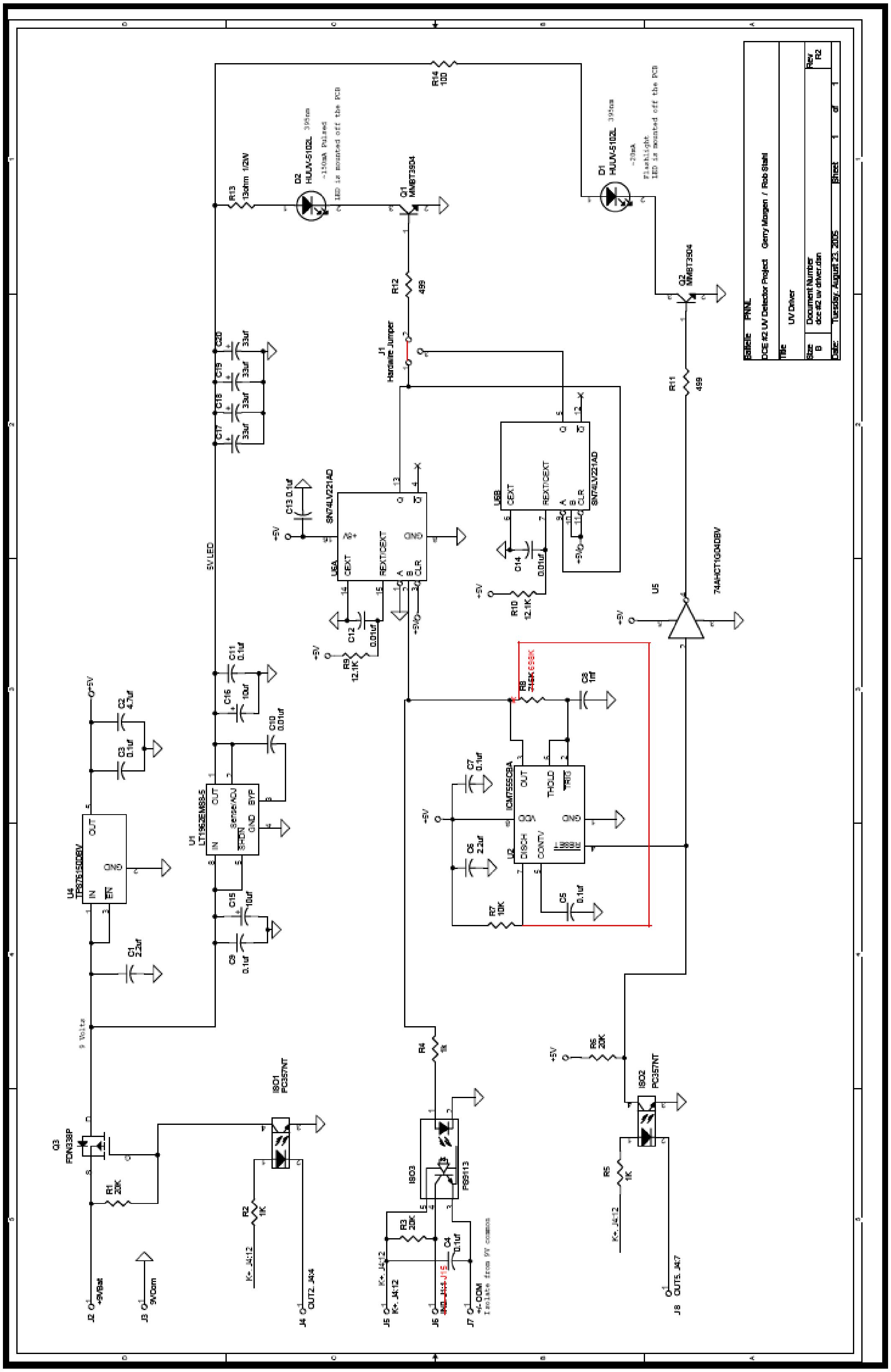

Figure 11. DCE UV photodiode driver.

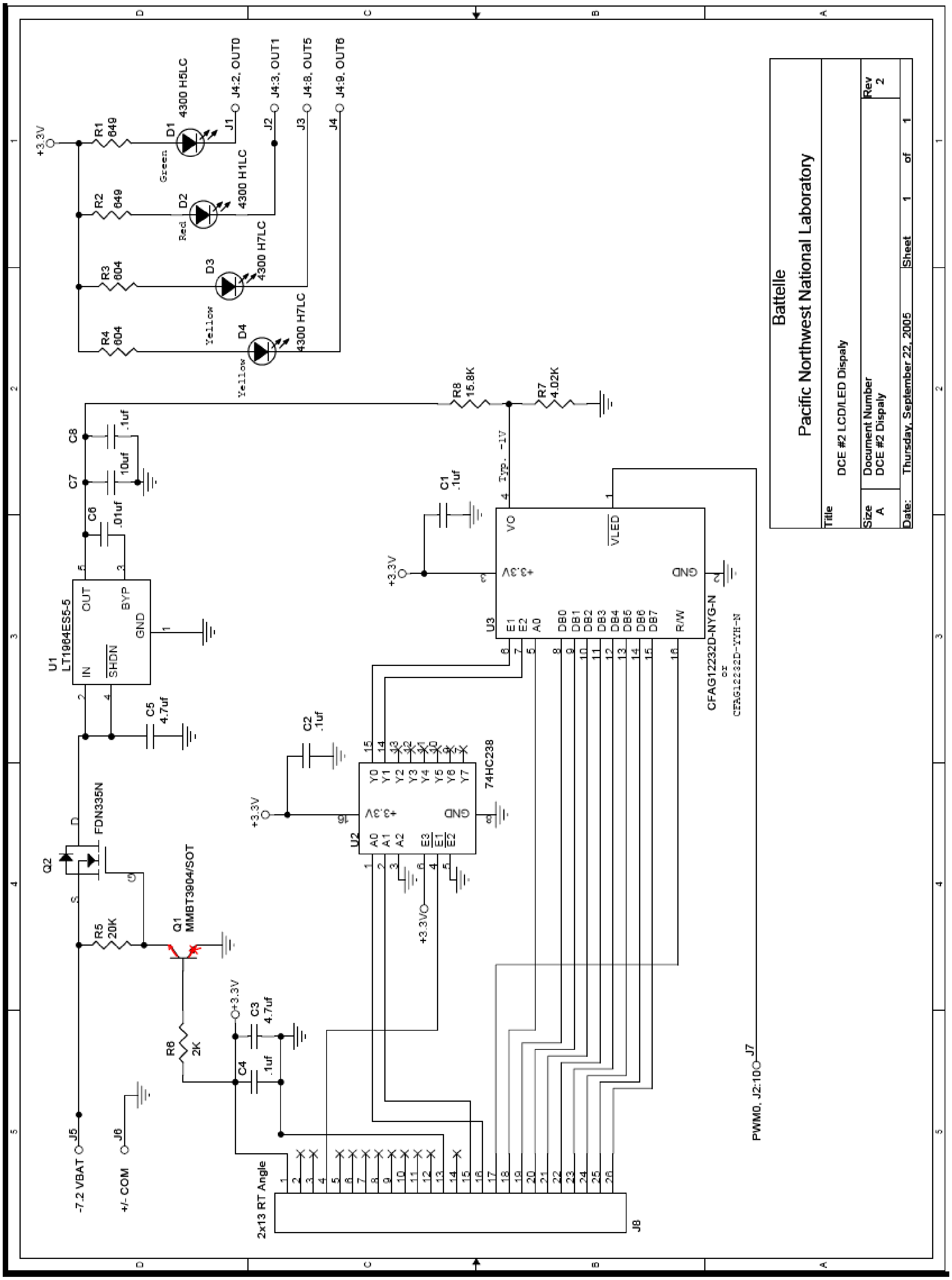

Figure 12. DCE display drivers.

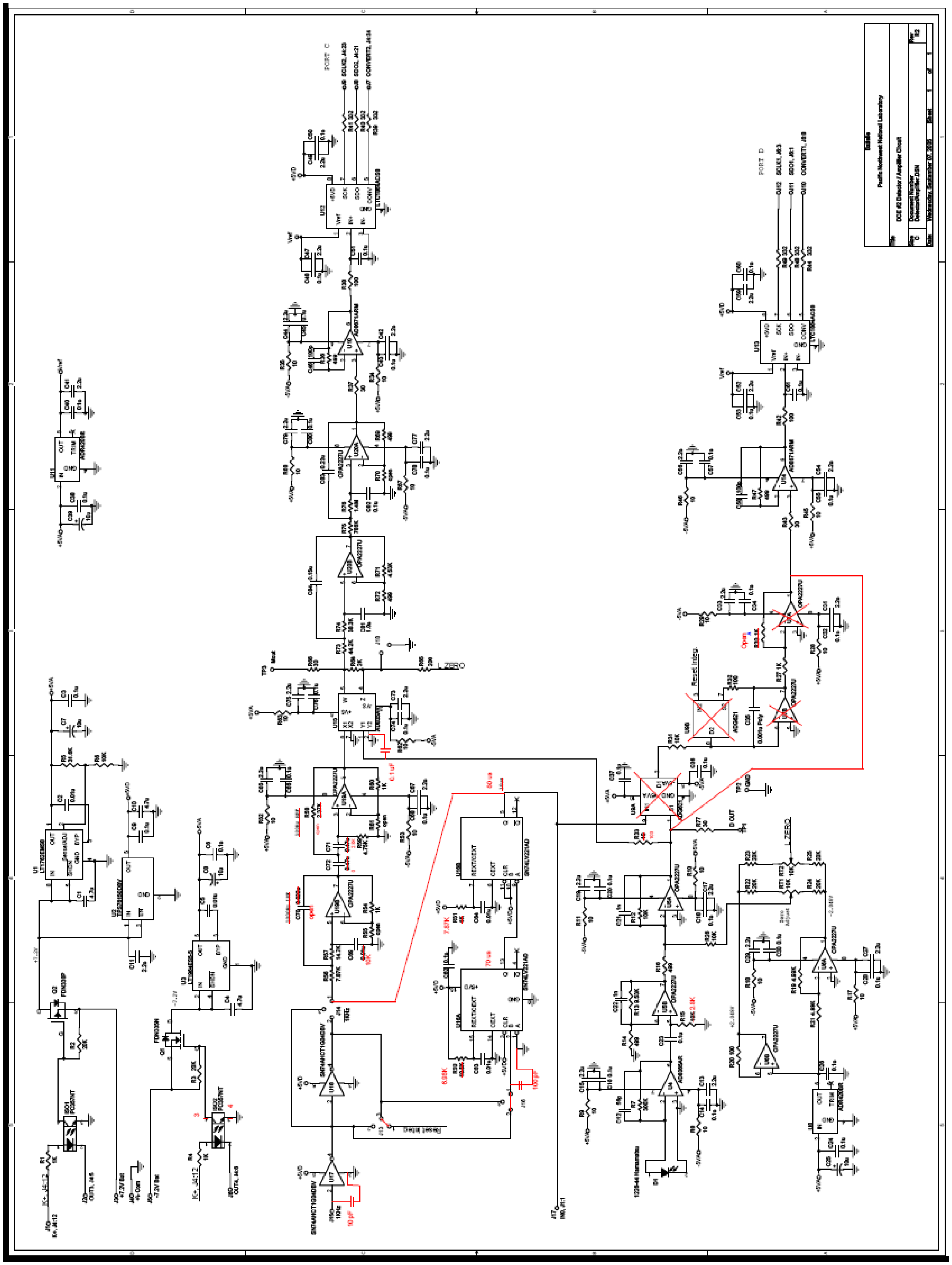

Figure 13. DCE optical signal amplifier schematics.

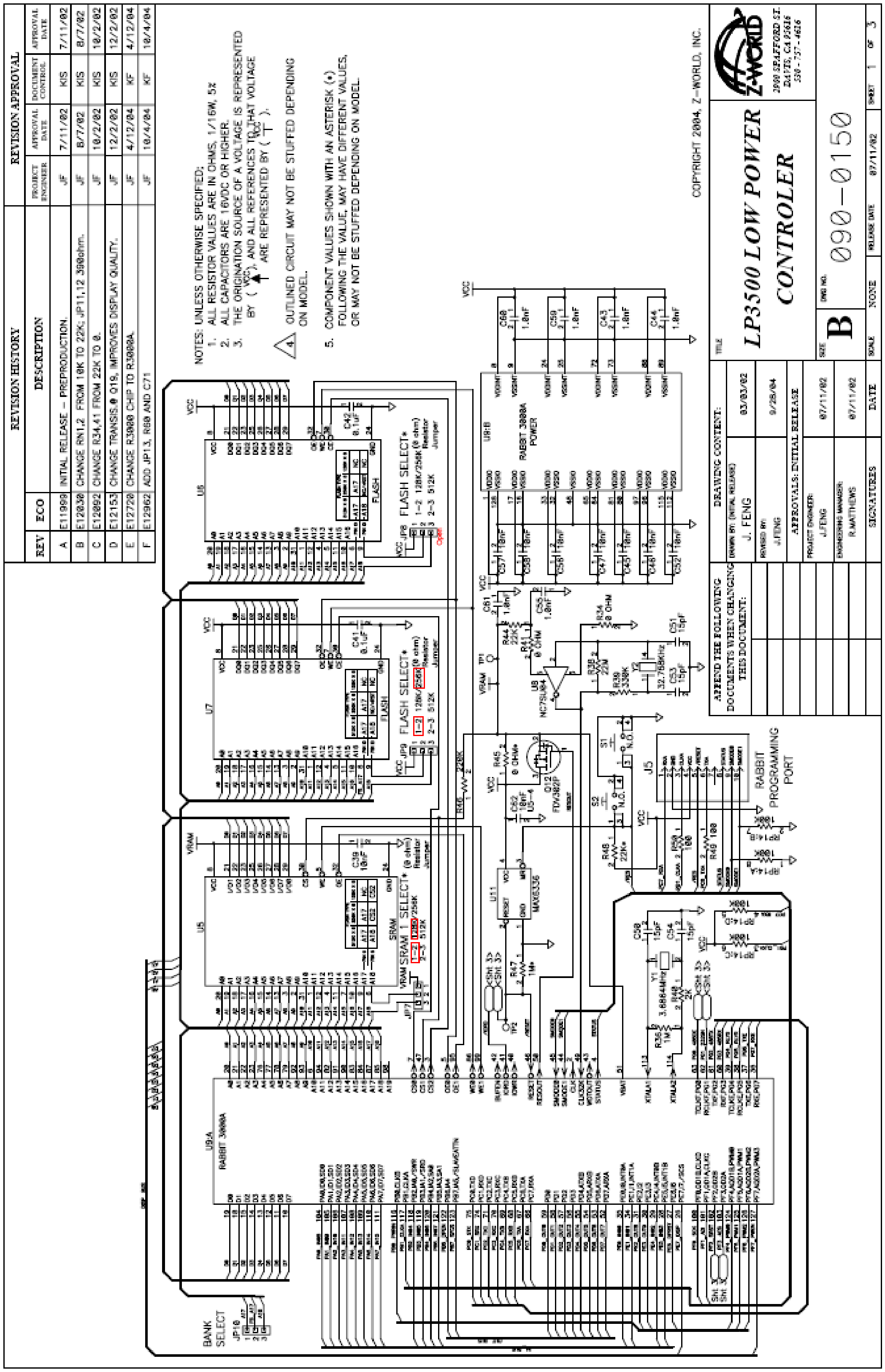

Figure 14. DCE microprocessor controller schematic.

Firmware Listing


```
//---------------------------------------------------------------------------
//                                                   -
/                                        DCE2MOD.c                    -
//                                                   -
//  Control and Data Acquisition program for detecting fluorencence of QDOTs   -
//  attached to Bio samples - Version #2 - with modifications to add control   -
//  switches [2 more], diodes in series with batteries and power supply inputs -
//  an input to define if running from Power Supply so no auto shutdown in     -
//  power supply mode of running                                   -
//                                                   -
//  Program runs on a Z-World LP3510 Fox controller board                   -
//  LCD and LED display is PNNL design using the LP3510 interface connector    -
//  UV LED excitation and Detector - Amp - A/D circuits are PNNL design        -
//                                                   -
//  * From Power-up, board initializes then latches the +V power on           -
//  * Pressing again powers the box down                             -
//        * Power On brings the controller to full power mode - display DCE #2   -
//  * On Green LED in switch is turned on                            -
//        * After short delay UV source and detection electronics turned on      -
//        * Measure switch toggles between Background and Measure sample modes  -
//  * Red LED in switch is turned on in Measure mode, off in BackGround mode   -
//  * Both Measure and BackGround modes display data that is being acquired    -
//  * FlashLight switch transitions to FlashLight mode from any other mode     -
//  * Must press FlashLight switch again to get back to BackGround - no others -
//  * Yellow LED in switch turns on - press again goes back to Measure ops     -
//  * Test switch similar to Measure - Yellow LED when pressed               -
//    * UV source pulse signal drives controller interrupt to acquire detector -
//        and synchronous detected signal = 1kHz rep rate - drive 10% duty cycle -
//        * Running average of 2048 detector and sync signal acquires displayed  -
//        * Display LEDs light based upon signal levels                  -
//                                                   -
//  * In battery power operations, unit shuts down after 15 minutes if        -
//  * no switch is pressed                                   -
//---------------------------------------------------------------------------
//
//---------------------------------------------------------------------------
//LP3510 Hardware Configuration and Connections                       -
//- On/Off Switch = In1, J1:2, PE1 [switch turns on OpToMOS + input low OFF] -
```

```
//- Measure Switch = In9, J1:11, PA1 [Vcc pullup switch takes to gnd]        -
//  - FlashLight Switch = IN10, J1:12, PA2 [Vcc pullup switch takes to gnd]    -
//  - Test Switch = IN11, J1:13, PA3 [Vcc pullup switch takes to gnd]          -
//- UV Drive Interrupt = In0, J1:1, PE0 [positive edge triggered interrupt]  -
//  - Power Source Input = In4, J1:5, PB2 [high = power supply - low = battery]-
//          - Green Status LED = Out0, J4:2, PD0 [high out-FET on to gnd-LED on]  -
//          - Red Status LED = Out1, J4:3, PD1 [high out-FET on to gnd-LED on]    -
//          - UV LED Drive Power = Out2, J4:4, PD2 [high output turns on power]   -
//          - Yellow1 Status LED = Out3, J4:5, PD3 [high out-FET on to gnd-LED on] -
//          - Yellow2 Status LED = Out4, J4:6, PD4 [high out-FET on to gnd-LED on]
//  - FlashLight = Out5, J4:7, PD5 [high out turns UVLED off & FlashLight on]  -
//          - Analog +/- Power = Out6, J4:8, PD6 [high turns on analog power]      -
//          - Latch Power On = Out7, J4:9, PD7 [high turns on FET latch main power] -
//  - FlashLight Switch Yellow LED = OUT8, J4:10, PE2 [turn on LED]            -
//  - Test Switch Yellow LED = OUT9, J4:11, PE3 [turn on LED]                  -
//  - LCD BackLight = PWM0, J2:10, PF4 [100 Hz 50% duty cycle]                 -
//  - On/Off Switch - Green LED = PWM1, J2:11, PF5 [100 Hz 50% duty cycle]     -
//  - Measure Switch - Red LED = PWM2, J2:12, PF6 [100 Hz 50% duty cycle]      -
//          - K+ jumpered to VCC of LP3510 - R10 & R12 = 0 ohms                -
//          - Detector signal acquired from serial A/D = LTC1864               -
//                    - Serial Port D operated in Clocked Synchronous mode      -
//                              - Receive Data = SRx, PC1, J6:1 [SDO on LTC1864]        -
//                              - Sync Clock = SCLK, PF0, J6:3 [SCK on LTC1864]         -
//                              - Convert/ChipSelect = SFCS, PB7, J6:8 [CONV on LTC1864]   -
//                              - Transmit Data = STx, PC0, J6:6 [SDI not used on LTC1864]  -
//          - Synchronous Detected Signal acquired from serial A/D = LTC1864            -
//          - Serial Port C operated in Clock Synchrouous mode - modify from RS232   -
//                    - Receive Data = RxC, PC3, J4:21 moved JP3,2-3 [SDO on LTC1864]        -
//                    - Sync Clock = TxE, J4:23, jump PF1 at U2pin17 to J4;23 [SCK LTC1864]  -
//                    - Convert/CS = RxE, J4:24, jump PF3 at U2pin23 to J2:24 [CONV LTC1864] -
//                    - Transmit Data = TxC, J4:20, PC2 moved JP4,2-3 [no SDI on LTC1864]    -
//          - Input Power = J2:16 is 7 to 10 Volts                                -
//          - Ground = J2:17 [+ J2:15, J2:13, J2:9 J1:9, J6:2, J4:1, J4:16, J4:25]     -
//-----------------------------------------------------------------------------
//
//-----------------------------------------------------------------------------

void  GetDataISR();

void  StatusLEDs(char  chVarSt);
void  SwitchLEDs(char  chVarSw);

int  unStatus;
static int  nReturnVal;

static unsigned int  unPortDIn;
static char  bPortDByteLo;
static char  bPortDByteHi;

static unsigned int  unPortCIn;
static char  bPortCByteLo;
```

```
static char  bPortCByteHi;

static char  bPowerOn;
static char  bBackGnd;
static char  bMeasure;
static char  bFlashLight;
static char  bTest;
static int  nDelayCount;
static int  nDisplayDLCounter;
static long int  lnActivityCounter;

static int  nPowerSwitch;
static int  nPowerSource;
static int  nFlashLightSwitch;
static int  nMeasureSwitch;
static int  nTestSwitch;
static char  bMeasureClick;
static char  bFlashLightClick;
static char  bTestClick;
static char  bMeasurePushed;
static char  bFlashLightPushed;
static char  bTestPushed;

static unsigned long int  freq;

static fontInfo fi6x8;
static windowFrame  wfLCDtext;

const char BitRevLUT[] =
{
0x00,0x80,0x40,0xC0,0x20,0xA0,0x60,0xE0,0x10,0x90,0x50,0xD0,0x30,0xB0,0x70,0xF0,
0x08,0x88,0x48,0xC8,0x28,0xA8,0x68,0xE8,0x18,0x98,0x58,0xD8,0x38,0xB8,0x78,0xF8,
0x04,0x84,0x44,0xC4,0x24,0xA4,0x64,0xE4,0x14,0x94,0x54,0xD4,0x34,0xB4,0x74,0xF4,
0x0C,0x8C,0x4C,0xCC,0x2C,0xAC,0x6C,0xEC,0x1C,0x9C,0x5C,0xDC,0x3C,0xBC,0x7C,0xF
C,
0x02,0x82,0x42,0xC2,0x22,0xA2,0x62,0xE2,0x12,0x92,0x52,0xD2,0x32,0xB2,0x72,0xF2,
0x0A,0x8A,0x4A,0xCA,0x2A,0xAA,0x6A,0xEA,0x1A,0x9A,0x5A,0xDA,0x3A,0xBA,0x7A,0x
FA,
0x06,0x86,0x46,0xC6,0x26,0xA6,0x66,0xE6,0x16,0x96,0x56,0xD6,0x36,0xB6,0x76,0xF6,
0x0E,0x8E,0x4E,0xCE,0x2E,0xAE,0x6E,0xEE,0x1E,0x9E,0x5E,0xDE,0x3E,0xBE,0x7E,0xFE,
0x01,0x81,0x41,0xC1,0x21,0xA1,0x61,0xE1,0x11,0x91,0x51,0xD1,0x31,0xB1,0x71,0xF1,
0x09,0x89,0x49,0xC9,0x29,0xA9,0x69,0xE9,0x19,0x99,0x59,0xD9,0x39,0xB9,0x79,0xF9,
0x05,0x85,0x45,0xC5,0x25,0xA5,0x65,0xE5,0x15,0x95,0x55,0xD5,0x35,0xB5,0x75,0xF5,
0x0D,0x8D,0x4D,0xCD,0x2D,0xAD,0x6D,0xED,0x1D,0x9D,0x5D,0xDD,0x3D,0xBD,0x7D,0x
FD,
0x03,0x83,0x43,0xC3,0x23,0xA3,0x63,0xE3,0x13,0x93,0x53,0xD3,0x33,0xB3,0x73,0xF3,
0x0B,0x8B,0x4B,0xCB,0x2B,0xAB,0x6B,0xEB,0x1B,0x9B,0x5B,0xDB,0x3B,0xBB,0x7B,0xF
B,
0x07,0x87,0x47,0xC7,0x27,0xA7,0x67,0xE7,0x17,0x97,0x57,0xD7,0x37,0xB7,0x77,0xF7,
0x0F,0x8F,0x4F,0xCF,0x2F,0xAF,0x6F,0xEF,0x1F,0x9F,0x5F,0xDF,0x3F,0xBF,0x7F,0xFF
};
```

```
static char  bNewData;
static int  nIndex;
static unsigned int  unDetectorAvg;
static unsigned int  unSyncAvg;
static unsigned int  unDetectorBack;
static unsigned int  unSyncBack;
static int  nDetectDiff;
static int  nSyncDiff;
static unsigned long int  ulnDetectorSum;
static unsigned long int  ulnSyncSum;
static unsigned int  unDetector[2048];
static unsigned int  unSync[2048];
static char  szSignalData[24];

//---------------------------------------------------------------------------
//  MAIN Program                                                            -
//---------------------------------------------------------------------------
nodebug
root void  main(void)
{
        int k;

        // Initialize the LP3510 board
        brdInit();

  // Shut down Watch Dog Timer - Virtual Driver shutdown in Program.Lib
  #asm
        ld      a,0x51
   ioi  ld      (WDTTR),a
   ld   a,0x54
   ioi  ld      (WDTTR),a
  #endasm

  // init OUT signals so no LEDs and electronic modules are powered
  digBankOut(0,0xff);                 // Out7 to Out0 high - not sinking
  digBankOut(1,0x00);                 // Out9 & Out8 low - not sourcing

  nIndex = 1;
  for(k=0;k<10000;k++)          //startup delay for power to stablize
  {
        nIndex = k;
  }

  nIndex = 0;
  // setup Serial Port D for sync serial operation
  BitWrPortI(PCFR,&PCFRShadow,1,0);         // PC0+PC1 alt serial Port D ops
  BitWrPortI(PFDDR,&PFDDRShadow,1,0);               // PF0 is output
  BitWrPortI(PFFR,&PFFRShadow,1,0);                         // PF0 is SCLKD
  // PB7-CSn via Reset default and brdInit() is an output set high
  // Setup Timer A7 to drive SCLKD
```

```
        WrPortI(TAT7R,&TAT7RShadow,1);              // 2nd fastest clock - approx 1MHz
  // Setup Port D serial receive registers - Control and Extended
        WrPortI(SDCR,&SDCRShadow,0x0c);  // no ops, recv enabled, clocked serial, no
interrupts
  WrPortI(SDER,&SDERShadow,0x00);            // no change - same as reset
  // start Timer A7
  BitWrPortI(TACSR,&TACSRShadow,1,0);

  // setup Serial Port C for sync serial opeation - hardware changes made to LP3510
  BitWrPortI(PCFR,&PCFRShadow,1,2);                          // PC2+PC3 alt serial Port C ops
  BitWrPortI(PFDDR,&PFDDRShadow,1,1);              // PF1 is output
  BitWrPortI(PFFR,&PFFRShadow,1,1);                          // PF1 is SCLKC
  BitWrPortI(PFDDR,&PFDDRShadow,1,3);              // PF3 is output - CSn for Port C serial
  BitWrPortI(PFDR,&PFDRShadow,1,3);                          // PF3 - CSn = high
  // Setup Timer A6 to drive SCLK
  WrPortI(TAT6R,&TAT6RShadow,1);                   // 2nd fastest clock - approx 1MHz
  // Setup Port C serial receive registers - Control and Extenced
  WrPortI(SCCR,&SCCRShadow,0x0c);// no ops, recv enabled, clocked serail, no interrupts
  WrPortI(SCER,&SCERShadow,0x00);                            // same as when reset
  // start Timer A6
  BitWrPortI(TACSR,&TACSRShadow,1,0);

  //  Setup Pulse Width Modulated Outputs
  freq = 100L;
  nReturnVal = pwmOutConfig(freq);             // 100 Hz frequency

  bPowerOn = 0;
        // Initialize Status
  bBackGnd = 0;
  bMeasure = 0;
  bFlashLight = 0;
  bTest = 0;
  bNewData = 0;
  bMeasureClick = 0;
  bFlashLightClick = 0;
  bTestClick = 0;
  bMeasurePushed = 0;
  bFlashLightPushed = 0;
  bTestPushed = 0;

  for(k=0; k<2048; k++)                                      // Initialize Data
  {
        unDetector[k] = 0;
   unSync[k] = 0;
  }
  unPortDIn = 0;
  unPortCIn = 0;
  unDetectorAvg = 0;
  unSyncAvg = 0;
  ulnDetectorSum = 0L;
  ulnSyncSum = 0L;
```

```
unDetectorBack = 0;
unSyncBack = 0;
nDelayCount = 0;
lnActivityCounter = 0L;
nDisplayDLCounter = 0;
nIndex = 0;

// Setup the text font on the LCD
        glXFontInit(&fi6x8,6,8,32,127,Font6x8);                    // 6x8 font

        // Setup text window to be the first 3 lines of display
        TextWindowFrame(&wfLCDtext,&fi6x8,1,0,121,24);

// Turn on the display and serial ports
nReturnVal = devPowerSet(DISPDEV|RS232DEV,1);
nReturnVal = powerMode(1);

        // Latch the Power Switch ON
digOut(7,0);                                   // Out7 low - sinks to gnd
bPowerOn = 1;

//  Turn on the Green LED in Power switch
pwmOut(1,0.5);                      // 50% duty cycle - maybe brighter?

//  Turn Red LED off in Measure switch - PWM outs init default high
pwmOut(2,0.0);

// Initialize the LCD Display
dispInit();

//  Turn off the LCD BackLight - if it has one
//  Set the output Low - init default is high
pwmOut(0,0.0);
//  BackLight will be on - off depending upon power source

// Setup Interrupt Vector
//SetVectExtern3000(0,GetDataISR);
nIndex = 0;
interrupt_vector ext0_intvec  GetDataISR;           // Separate I & D Space

// write ID Banner to LCD line 1
TextGotoXY(&wfLCDtext,0,0);
        TextPrintf(&wfLCDtext," USAF DCE Number 2 ");

// wait some before turning on electronics
        nDelayCount = 0;
for(k=0;k<10000;k++)
{
 nDelayCount++;
}
        bBackGnd = 1;
```

```
digOut(2,0);                                        // UVLED driver power
        digOut(6,0);                                // Analog Power

TextGotoXY(&wfLCDtext,0,1);
TextPrintf(&wfLCDtext,"   BackGround Mode  ");
WrPortI(I0CR,&I0CRShadow,0x09);             // enable INT 0 positive edge

// more delay time to get finger off the switch
for(k=0;k<50;k++)
{
 TextGotoXY(&wfLCDtext,0,1);
         TextPrintf(&wfLCDtext,"   BackGround Mode  ");
}

while(1)
{
        // read inputs
 nPowerSwitch = digIn(1);                   // low when pushed active
 nPowerSource = digIn(4);           // high = power supply - low = battery
 nMeasureSwitch = digIn(9);                 // low when pushed active
 nFlashLightSwitch = digIn(10);             // low when pushed active
 nTestSwitch = digIn(11);                   // low when pushed active


 // check power switch
 if(nPowerSwitch == 0)
 {
        digOut(7,1);                        // THIS SHUTS DOWN THE POWER !!!!!!!
 }

        // process battery or power supply operation
 if(nPowerSource == 0)
 {
  pwmOut(0,0.0);                    // turn off backlight to save battery power

        lnActivityCounter++;
        if(lnActivityCounter > 1200000L)                // around 15 minutes
        {
                digOut(7,1);                // THIS SHUTS DOWN THE POWER !!!!!
        }
 }
 else
 {
                        pwmOut(0,0.5);                  // turn on backlight
 }

 // process switch input data - switch press and release = pushed
 if(nMeasureSwitch == 0)
        bMeasureClick = 1;
 else
 {
```

```
        if(bMeasureClick)
  {
        bMeasurePushed = 1;
   bMeasureClick = 0;
  }
}

if(nFlashLightSwitch == 0)
        bFlashLightClick = 1;
else
{
        if(bFlashLightClick)
  {
        bFlashLightPushed = 1;
   bFlashLightClick = 0;
  }
}

if(nTestSwitch == 0)
        bTestClick = 1;
else
{
        if(bTestClick)
  {
        bTestPushed = 1;
   bTestClick = 0;
  }
}

                    // check if Measure Switch is pushed
if(bMeasurePushed)
{
        lnActivityCounter = 0L;                                   // reset activity counter
 bMeasurePushed = 0;
                              if(!bFlashLight)
  {
        if(bBackGnd)                                    // go to Measure Mode
        {
                                                  bBackGnd = 0;
                    bMeasure = 1;
    bFlashLight = 0;
    bTest = 0;

        SwitchLEDs('M');                                // Turn ON Measure Red LED

        unDetectorBack = unDetectorAvg;
        unSyncBack = unSyncAvg;

        // write MODE:  Measure
        TextGotoXY(&wfLCDtext,0,1);
        TextPrintf(&wfLCDtext,"    Measure Mode    ");
```

```
        }
        else if(bTest || bMeasure)          // go to BackGround Mode
        {
                bBackGnd = 1;
        bMeasure = 0;
      bFlashLight = 0;
      bTest = 0;

        SwitchLEDs('N');                                    // Turn all off

        // write MODE:  Backgnd
        TextGotoXY(&wfLCDtext,0,1);
        TextPrintf(&wfLCDtext,"  BackGround Mode   ");
        }
  }
}

// check if FlashLight Switch is pushed
if(bFlashLightPushed)
{
        lnActivityCounter = 0L;                 // reset activity counter
  bFlashLightPushed = 0;

        if(!bFlashLight)                        // go to flashlight ops
  {
                                bBackGnd = 0;
                bMeasure = 0;
   bFlashLight = 1;
   bTest = 0;
   SwitchLEDs('F');                   // turn ON Flashlight yellow LED
   digOut(5,0);                       // flashlight on + stop pulsing UVLED
                          // write MODE: FlashLigh
   TextGotoXY(&wfLCDtext,0,1);
   TextPrintf(&wfLCDtext,"  Flash Light Mode  ");

           WrPortI(I0CR,&I0CRShadow,0x00);                        // disable INT 0
  }
  else if(bFlashLight)                //stop flashlight - go to background
  {
                                bBackGnd = 1;
                bMeasure = 0;
   bFlashLight = 0;
   bTest = 0;

   digOut(5,1);                                 // flashlight off - pulsing UVLED on

        SwitchLEDs('N');                                        // Turn all off

   // write MODE:  Backgnd
   TextGotoXY(&wfLCDtext,0,1);
   TextPrintf(&wfLCDtext,"  BackGround Mode   ");
```

```
            WrPortI(I0CR,&I0CRShadow,0x09);                    // enable INT 0 positive edge
  }
}

// check it Test Switch is pushed
if(bTestPushed)
{
        lnActivityCounter = 0L;                         // reset activity counter
  bTestPushed = 0;

        if(!bFlashLight)
  {
        if(bBackGnd)                                                    // go to test ops
        {
                                                bBackGnd = 0;
                    bMeasure = 0;
      bFlashLight = 0;
      bTest = 1;

        unDetectorBack = unDetectorAvg;
        unSyncBack = unSyncAvg;

        SwitchLEDs('T');                                // Turn ON test Yellow LED

        // write MODE: Test
        TextGotoXY(&wfLCDtext,0,1);
        TextPrintf(&wfLCDtext,"      Test Mode     ");

        }
        else if(bTest || bMeasure)                              // stop test - go to background
        {
                                                bBackGnd = 1;
                    bMeasure = 0;
      bFlashLight = 0;
      bTest = 0;

        SwitchLEDs('N');                                                // all LEDs off

        // write MODE:  Backgnd
        TextGotoXY(&wfLCDtext,0,1);
        TextPrintf(&wfLCDtext,"  BackGround Mode   ");
        }
  }
}

// check if display numbers and status LEds should be updated
if(nDisplayDLCounter >= 640)
{
        nDisplayDLCounter = 0;
```

```
        // check what data to displa
                if(!bFlashLight)// Display detector and sync signal data
        {
          WrPortI(I0CR,&I0CRShadow,0x00);                          // disable INT 0
        unDetectorAvg = (unsigned int)(ulnDetectorSum >> 11);      // div by 2048
       unSyncAvg = (unsigned int)(ulnSyncSum >> 11);
          WrPortI(I0CR,&I0CRShadow,0x09);                   // enable INT 0 positive edge
}

       // format Signal data for display
      if(bFlashLight)
{
      strcpy(szSignalData,"                 ");
}
else if(bMeasure || bTest)
         {
                nDetectDiff = (int)(unDetectorAvg - unDetectorBack);
                nSyncDiff = (int)(unSyncAvg - unSyncBack);
  if(bMeasure)
  {
      if(nDetectDiff < 0)
      nDetectDiff = 0;
    if(nSyncDiff < 0)
      nSyncDiff = 0;
  }

      nReturnVal = sprintf(szSignalData,"D=%+06d  L=%+06d",nDetectDiff,nSyncDiff);
}
                          else
      {
      nReturnVal = sprintf(szSignalData,"D=%05u   L=%05u ",unDetectorAvg,unSyncAvg);
         }
        // display Signal data on LCD
       TextGotoXY(&wfLCDtext,1,2);
      TextPrintf(&wfLCDtext,"%s\n",szSignalData);

// update display LEDs
if(bMeasure)
{
      if( (nDetectDiff > 250) && (nSyncDiff > 2000) )
          {
      StatusLEDs('R');                                   // Red LED
          }
        else if( (nDetectDiff > 75) && (nSyncDiff > 600) )
       {
      StatusLEDs('W');                                   // Both Yellow LEds
       }
  else if( (nDetectDiff > 25) && (nSyncDiff > 200) )
      {
      StatusLEDs('Y');                                   // Left Yellow LED
  }
```

```
            else if( (nDetectDiff < -30) && (nSyncDiff < -300) )
            {
        StatusLEDs('F');                                // All LEDs
            }
            else
            {
        StatusLEDs('G');                                // Green LED
            }

            }
    else
    {
        StatusLEDs('N');                                        // All LEDs off
    }

           }
   else
   {
        nDisplayDLCounter++;                            // increment counter
   }

  }      // end of While Loop

}                // end of Main

//----------------------------------------------------------------------------
//       Interrupt Routine to get data from serial A/D                        -
//----------------------------------------------------------------------------

nodebug root interrupt void  GetDataISR()
{
 #asm
                nop
        ;trigger A/D for new sample detector signal
   nop
   nop
   nop
   nop
   ld     a,(PBDRShadow)
   or     0x80
   ld     (PBDRShadow),a
   ioi    ld        (PBDR),a                         ;PB7-CSnD = high
   nop                                              ;samples detector signal
 #endasm

 // process last detector data value
 ulnDetectorSum -= unDetector[nIndex];
 ulnDetectorSum += unPortDIn;
 unDetector[nIndex] = unPortDIn;

 // process last synchrouous data value
```

```
ulnSyncSum -= unSync[nIndex];
ulnSyncSum += unPortCIn;
unSync[nIndex] = unPortCIn;

        // Check data array indexing
nIndex++;
if(nIndex >= 2048)
        nIndex = 0;

#asm
 nop                            ;trigger A/D to sample synchronous signal
 ld     a,(PFDRShadow)
 or     0x08
 ld     (PFDRShadow),a
 ioi    ld      (PFDR),a                ;PF3-CSnC = high
 nop                                    ;samples synchrouous signal
 nop
#endasm

        // Read new detector data value
// Get 2 bytes = 16 bits via serial Port D
#asm
 ld     bc,BitRevLUT            ;setup bc for later data acquisition
        ld      a,(PBDRShadow)
 and    0x7f
 ld     (PBDRShadow),a
 ioi    ld      (PBDR),a                ;PB7-CSnD = low
 ld     a,0x4c
 ioi    ld      (SDCR),a                ;start receive byte
                D1Loop:                         ;wait for data to shift in
 ioi    ld      a,(SDSR)
 bit    7,a
 jp     z,D1Loop
        ioi     ld      a,(SDAR)        ;read data [high byte] + start again
 ld     h,0x00
 ld     l,a
 add    hl,bc                           ;bit reverse of byte data - LUT
 ld     a,(hl)
        ld      (bPortDByteHi),a
                D2Loop:                         ;wait for data to shift in
 ioi    ld      a,(SDSR)
 bit    7,a
 jp     z,D2Loop
 ioi    ld      a,(SDDR)                ;read 2nd byte [low byte]
 ld     h,0x00
 ld     l,a
 add    hl,bc                           ;bit reverse of byte data - LUT
 ld     a,(hl)
 ld     (bPortDByteLo),a
 ld     l,a
 ld     a,(bPortDByteHi)
```

```
 ld     h,a
 ld     (unPortDIn),hl                      ;16-bit data stored
#endasm

// Read new synchronous data value
        // Get 2 Bytes = 16 bits via serial Port C
#asm
 ld     bc,BitRevLUT                        ;setup bc for later data acquisition
 ld     a,(PFDRShadow)
 and    0xf7
 ld     (PFDRShadow),a
 ioi    ld      (PFDR),a                            ;PF3-CSnC = low
 ld     a,0x4c
 ioi    ld      (SCCR),a                            ;start receive byte
                C1Loop:                                     ;wait for data to shift in
 ioi    ld      a,(SCSR)
 bit    7,a
 jp     z,C1Loop
        ioi     ld      a,(SCAR)                    ;read data [high byte] + start again
 ld     h,0x00
 ld     l,a
 add    hl,bc                                       ;bit reverse of byte data - LUT
 ld     a,(hl)
        ld      (bPortCByteHi),a
                C2Loop:                                     ;wait for data to shift in
 ioi    ld      a,(SCSR)
 bit    7,a
 jp     z,C2Loop
 ioi    ld      a,(SCDR)                            ;read 2nd byte [low byte]
 ld     h,0x00
 ld     l,a
 add    hl,bc                                       ;bit reverse of byte data - LUT

 ld     a,(hl)
 ld     (bPortCByteLo),a
                ld      l,a
 ld     a,(bPortCByteHi)
 ld     h,a
 ld     (unPortCIn),hl              ;16-bit data stored
#endasm
if(unPortDIn < 256)
{
        if(nIndex == 0)
        unPortDIn = unDetector[2047];
 else
        unPortDIn = unDetector[nIndex-1];
}
if(unPortCIn < 256)
{
        if(nIndex == 0)
        unPortCIn = unSync[2047];
```

```
   else
         unPortCIn = unSync[nIndex-1];
  }
}

//--------------------------------------------------------------------------
//  Status LEDs Routine                                         -
//  4 LEDs - Green, Yellow1, Yellow2, and Red                           -
//  Routine passed a char that specifies which LEDs are on                -
//  char N = None                                            -
//  char G = Green LED                                          -
//  char Y = Yellow1 LED [left one]                                -
//  char W = Yellow1 and Yellow2 LEDs                                -
//  char R = Red LED                                          -
//  char F = All LEDs [used to indicate a failure]                  -
//  default will be None                                        -
//--------------------------------------------------------------------------

nodebug
root void  StatusLEDs(char  chVarSt)
{
         switch(chVarSt)
  {
         case 'G':
                         digOut(0,0);                    // Green = On
                         digOut(1,1);                    // Red = Off
                         digOut(3,1);                    // Yellow1 = Off
                         digOut(4,1);                    // Yellow2 = Off
   break;
   case 'R':
                         digOut(0,1);                    // Green = Off
                         digOut(1,0);                    // Red = On
                         digOut(3,1);                    // Yellow1 = Off
                         digOut(4,1);                    // Yellow2 = Off
   break;
   case 'Y':
                         digOut(0,1);                    // Green = Of
                         digOut(1,1);                    // Red = Off
                         digOut(3,0);                    // Yellow1 = On
                         digOut(4,1);                    // Yellow2 = Off
   break;
   case 'W':
                         digOut(0,1);                    // Green = Off
                         digOut(1,1);                    // Red = Off
                         digOut(3,0);                    // Yellow1 = On
                         digOut(4,0);                    // Yellow2 = On
   break;
   case 'F':
                         digOut(0,0);                    // Green = On
                         digOut(1,0);                    // Red = On
                         digOut(3,0);                    // Yellow1 = On
```

```
                        digOut(4,0);                    // Yellow2 = On
   break;
   case 'N':
   default:
                        digOut(0,1);                    // Green = Off
                        digOut(1,1);                    // Red = Off
                        digOut(3,1);                    // Yellow1 = Off
                        digOut(4,1);                    // Yellow2 = Off
   break;
  }
}
//---------------------------------------------------------------------------
//  Switch LEDs Routine                                                     -
//  3 LEDs - Yellow, Yellow, and Red                                        -
//  Routine passed a char that specifies which LEDs are on                  -
//  char N = None                                                           -
//  char T = Test Switch Yellow LED - uses digital output 9                 -
//  char F = Flash Light Switch Yellow LED - uses digital output 8          -
//  char M = Measure Switch Red LED - uses PWM output 2                     -
//  default will be None                                                    -
//---------------------------------------------------------------------------
nodebug
root void  SwitchLEDs(char  chVarSw)
{
        switch(chVarSw)
  {
        case 'M':
                        pwmOut(2,0.5);          // Measure Red = On - 50% duty cycle
                        digOut(8,0);                    // Flash Yellow = Off
                        digOut(9,0);                    // Test Yellow = Off
   break;

   case 'F':
                        pwmOut(2,0.0);          // Measure Red = Off
                        digOut(8,1);                    // Flash Yellow = On
                        digOut(9,0);                    // Test Yellow = Off
   break;
   case 'T':
                        pwmOut(2,0.0);          // Measure Red = Off
                        digOut(8,0);                    // Flash Yellow = Off
                        digOut(9,1);                            // Test Yellow = On
   break;
   case 'N':
   default:
                        pwmOut(2,0.0);          // Measure Red = Off
                        digOut(8,0);                    // Flash Yellow = Off
                        digOut(9,0);                    // Test Yellow = Off
   break;
  }
}
```

Cartridge Design

The concentrating cartridge design is driven by the following considerations:

- Large enough to hold comfortably with gloved hands
- Vacuum sealed for ease of filling via syringe through a rubberized septum
- Able to accommodate and concentrate at 500 micoliter sample volume
- Integral magnet to allow filling and concentration in the absence of the DCE instrument
- Rugged enough to resist drop fracture or separation
- Stable enough to withstand external decontamination via bleach solution without compromising the contents.
- Long term sealed for chain of custody considerations.

Photographs of the assembled cartridges are shown in Figure 15. The photographs show close up views of the concentration magnet and the fill septum.

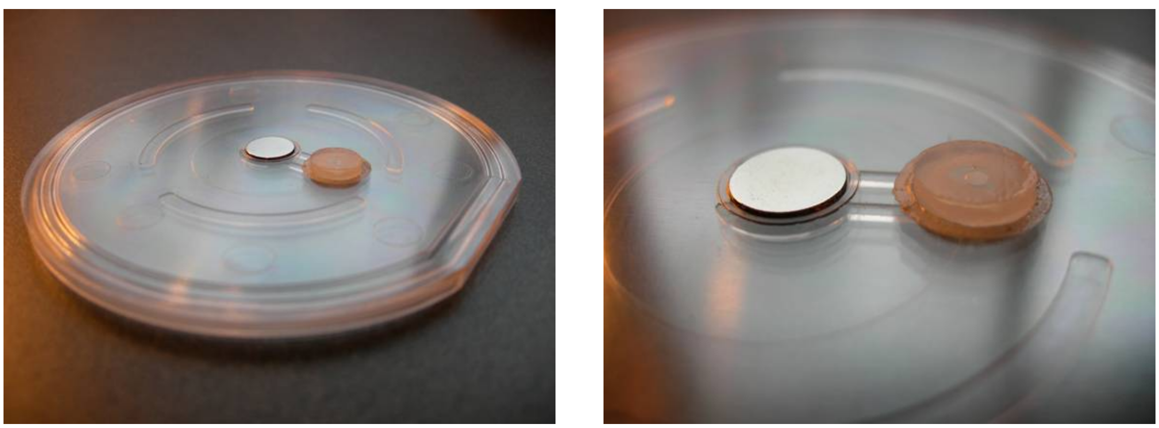

Figure 15. Photographs of assembled DCE cartridge including close up of magnet and injection septum.

Dimensioned drawings of the cartridge assembly are shown in Figures 16-18. The main body of the cartridge is an injection molded transparent polycarbonate construction. The main body is ultrasonically welded to a cover sheet of clear polycarbonate forming a 500 micoliter cavity with a guided fluid channel so that all material injected flows past the embedded magnet.

The two piece cartridge bodies are molded and ultrasonic welded by Plastic Injection Molding, Inc (PMI), Richland, WA 509-375-4260. The molds and the welding tooling have been retained in storage by PMI for future fabrication or client disposition at a later date.

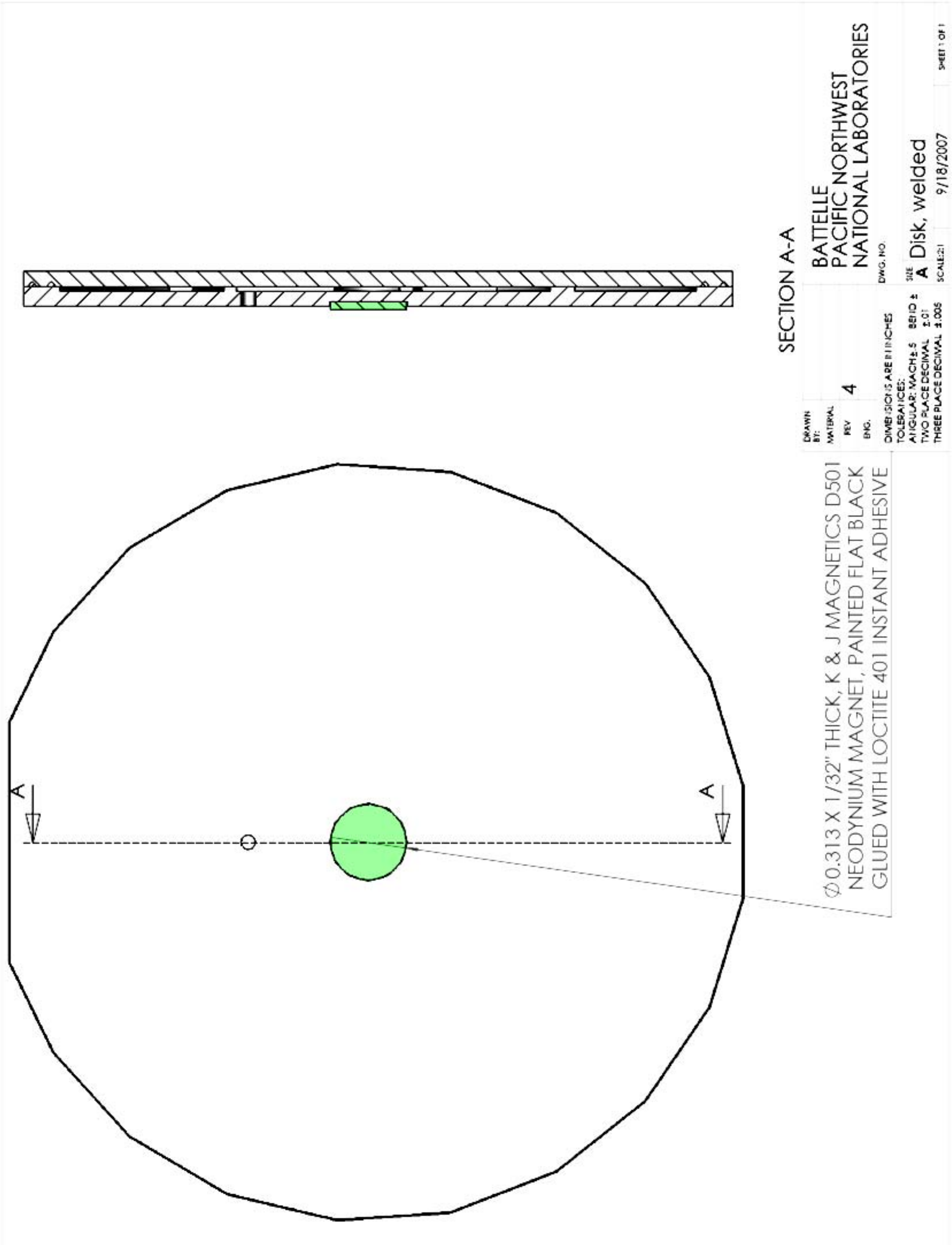

Figure 16. Dimensional drawing of DCE cartridge assembly.

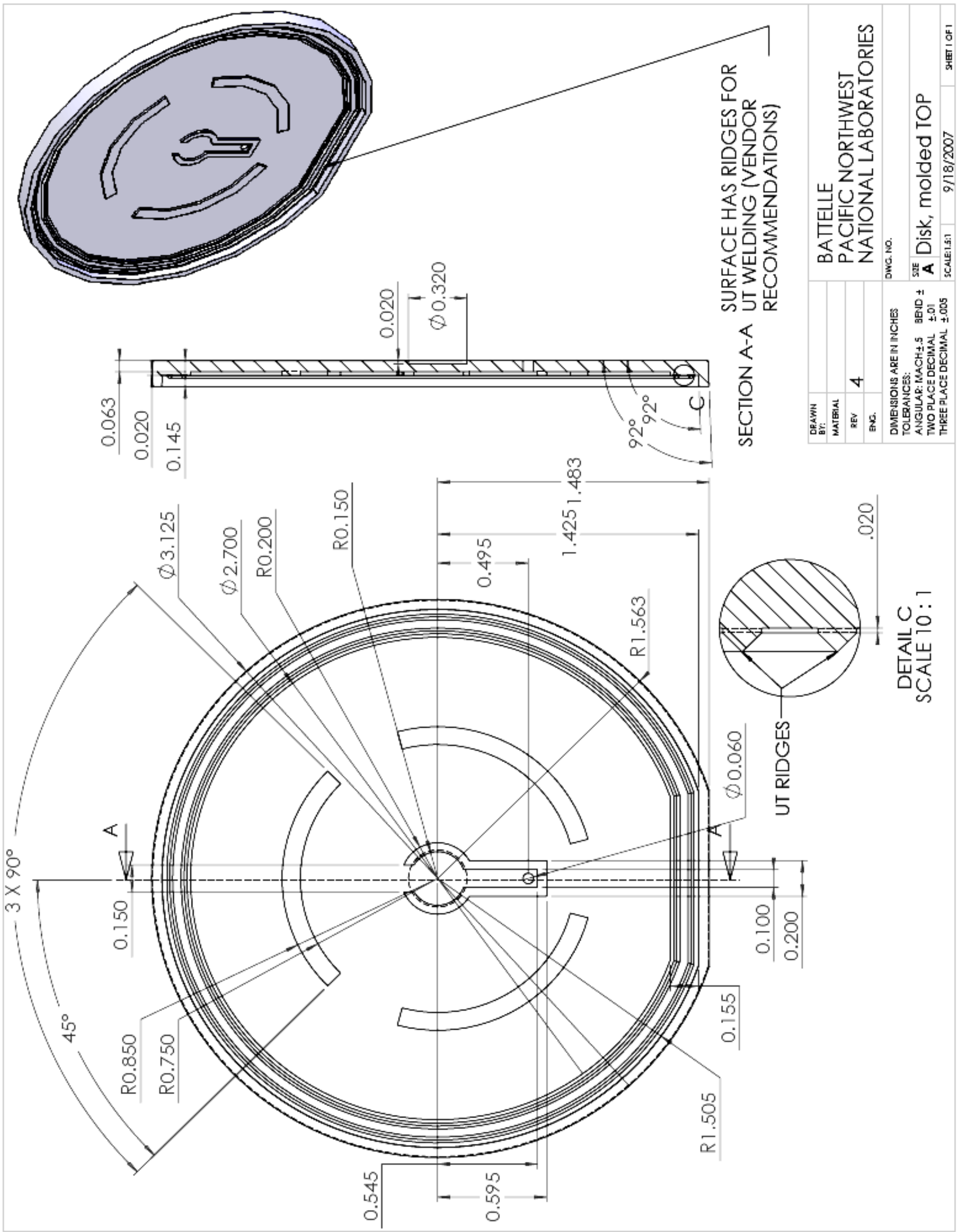

Figure 17. Dimensional drawing of DCE cartridge molded top detail.

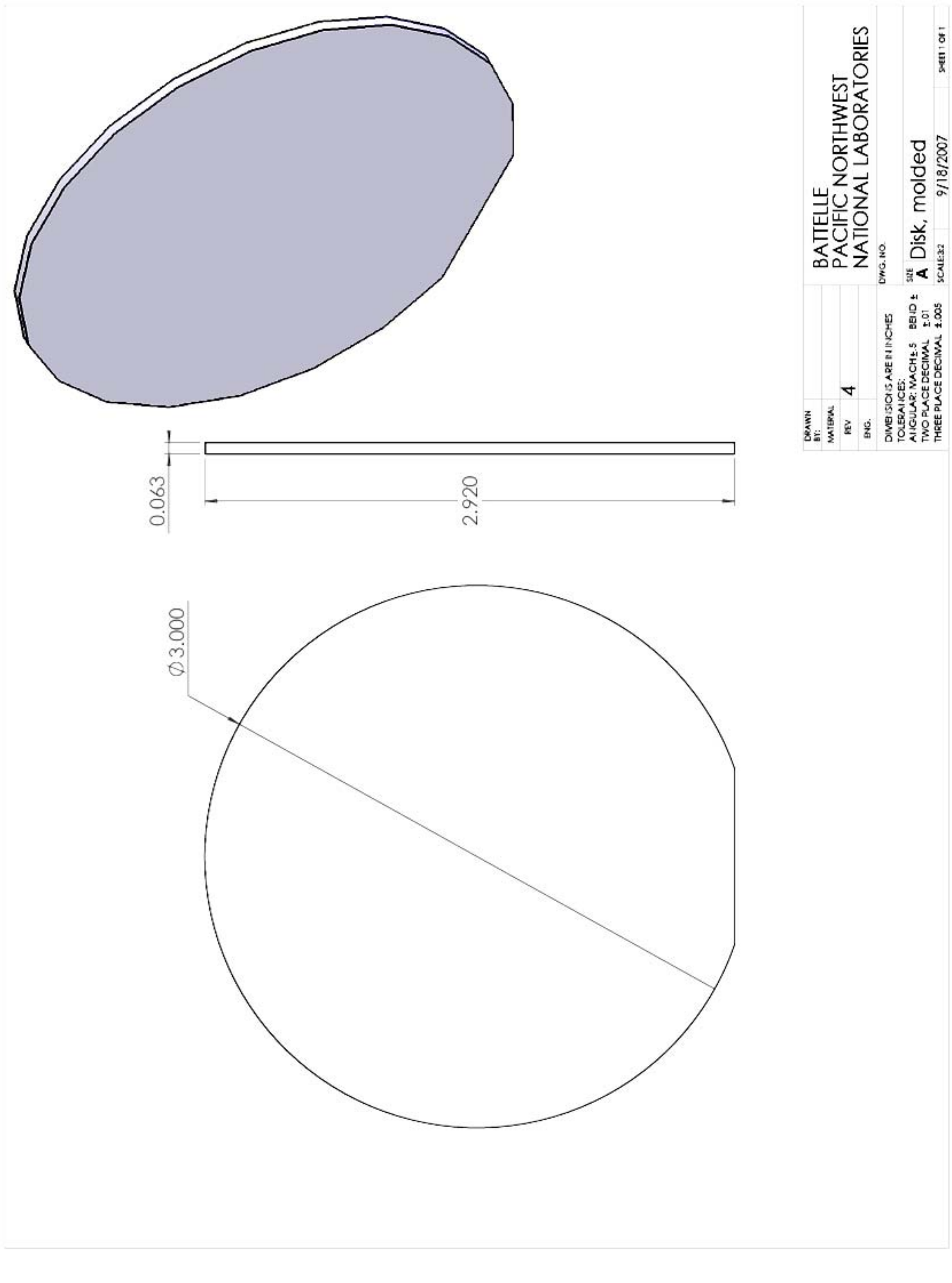

Figure 18. DCE cartridge bottom detail.

Process notes on cartridge assembly:

The cartridges are outgassed by overnight baking at 95C and 20" Hg vacuum. The prototype were outgases at PNNL.
Prepainted (black) 0.313x0.0313 neodymium disk magnets (D501, K&J Magnetics) are affixed to the exterior center recess using Loctite 401 adhesive. These magnet disks attract and concentrate the magnetic beads for the optical assay when the alalyte stream is injected into the cartridge.
8.0 mm diameter injection port septa are cut from sheet stock of .050" - 50 durometer silicone bonded to .002" FEP Teflon. (part ST58/2, Specialty Silicone, 518-885-8826). The dual material provides a lower air diffusion profile.
The septa are affixed to the cartridge by applying a ring of polyurethane adhesive sealant (3M Scotch Seal 560 Gray) concentric with the cartridge fill port, then affixing each septum with the Teflon side up.
After >2 hours cure time the cartridges are evacuated by puncturing each septum with a 30 gauge needle plumbed to 22" Hg vacuum. In the prototype production run about 10% seal failure occurred after seal puncture via leakage through the cartridge perimeter.
Groups of four cartridges oriented back-to-back are immediately sealed in bags using a Food Saver V2220 vacuum sealer for distribution and shipment.

Notes on cartridge testing:

Limited initial testing was performed on the assembled cartridges. After 25 days two bags were opened to test vacuum and the water draw capacity of the eight cartridges. The vacuum in the bags at this time was greater 10.3" Hg. Actual vacuum levels are likely higher than reported by 1 to 2 in Hg as the measurement process injects a small amount of air at atmospheric pressure before each measurement.

Water draw tests were conducted by filling a 3 ml syringe with a 30 gauge needle to 900 µl with no plunger in the syringe. The open syringe was used to puncture the septum and 5 minutes was allowed for flow, then the volume change at the syringe was noted. Final flows varied from 350 – 900 µl and took between 22 and five minutes depending the insertion technique.

Because of the insertion variability, a simple puncture jig would make an excellent accessory to insure minimum initial evacuation failure and maximum low and minimum fill times during cartridge operation.