MGL

FIRMWARE DESIGN SPECIFICATION

May 22, 2003

CANDELA CORPORATION
530 BOSTON POST ROAD
WAYLAND, MASSACHUSETTS 01778-1883

TABLE OF CONTENTS

1.0 PURPOSE

The purpose of this document is to describe the function and architecture of the MGL laser firmware.

The intended audience for this document include the following:

- Marketing
- Engineering
- Regulatory Affairs
- Clinical Research
- Quality Assurance
- Field Service
- Manufacturing

2.0 DEFINITIONS

The following are definitions of terms and acronyms used in this document:

ADC:	On-chip Analog-Digital Converter	PFN:	Pulse Forming Network
CDRH:	Center for Devices and Radiological Health	PWM:	Pulse Width Modulation
CP:	Cal Port Energy	RAM:	Random Access Memory
CPU:	Central Processing Unit (µ-processor)	ROM:	Read Only Memory
D/A:	Digital-to-analog Converter	SCI:	Serial Communication Interface
DCD:	Dynamic Cooling Device	SPI:	Serial Peripheral Interface
DI:	De-Ionized (Water)	TSDP:	Touch Screen Display Panel (display with touch panel overlay)
HD:	Head Energy	Tx:	Transmission (CP/HD)
HVPS:	High Voltage Power Supply	UI:	User Interface
I/O:	Input/Output	xHD:	Expected Head Energy
LSB:	Least significant bit		
PCB:	Printed Circuit Board		

3.0 REFERENCES

The following reference documents were used in the development of this specification:

- 8502-30-0010 CANDELA Firmware Standard Operating Procedure (FSOP)
- 9914-90-0880 MGL Engineering Design Specification
- 1010-06-3020 MGL Firmware Requirements Specification (FRS)

4.0 CONTROL SYSTEM HARDWARE

4.1. Hardware Description

The Laser control system consists of a system controller, Flash-Lamp, HVPS, PFN, optics, energy detectors, display with touch screen interface, and a controller electronics. The system controller section of the PCB is comprised of a 68HC12 CPU, embedded peripheral controllers, and the firmware to operate the laser. The CPU interfaces with the inputs and outputs of the various control subsystems.

The capability of the system is enhanced by a four channel 12-bit A/D and a single channel 12-bit D/A converter. Both of these devices interface with the CPU via a SPI bus. Another CPU I/O port is configured as an eight channel 8-bit A/D providing for additional low resolution analog inputs.

The following user interface components are monitored or controlled by the firmware: Touch Screen Display Panel (TSDP), READY lamp, Aiming Beam, audible alarm, and the footswitch/fingerswitch. The TSDP and the diagnostic port are described below:

4.1.1. Touch Screen Display Panel(TSDP)

The Touch Screen Display Panel is composed of a Liquid Crystal display with a touch panel overlay The firmware displays a mixture of graphical icons and text to convey system settings and status to the user. Additionally, the firmware is also responsible for monitoring and processing user input from the touch panel.

4.1.2. Ready Lamp

The firmware turns this lamp on whenever the system is in the READY state. The lamp serves as a visual indicator that the system is in the READY state.

4.1.3. Aiming Beam

The aiming beam serves as an indicator of the position where the laser will pulse. The aiming beam is also used as a remote READY state indicator. The aiming beam intensity is set from the Option Menu and is controlled by PWM of the diode.

4.1.4. Audible Alarm

The firmware uses an audible alarm to generate beep tones which alert the user that one of the following conditions has occurred:

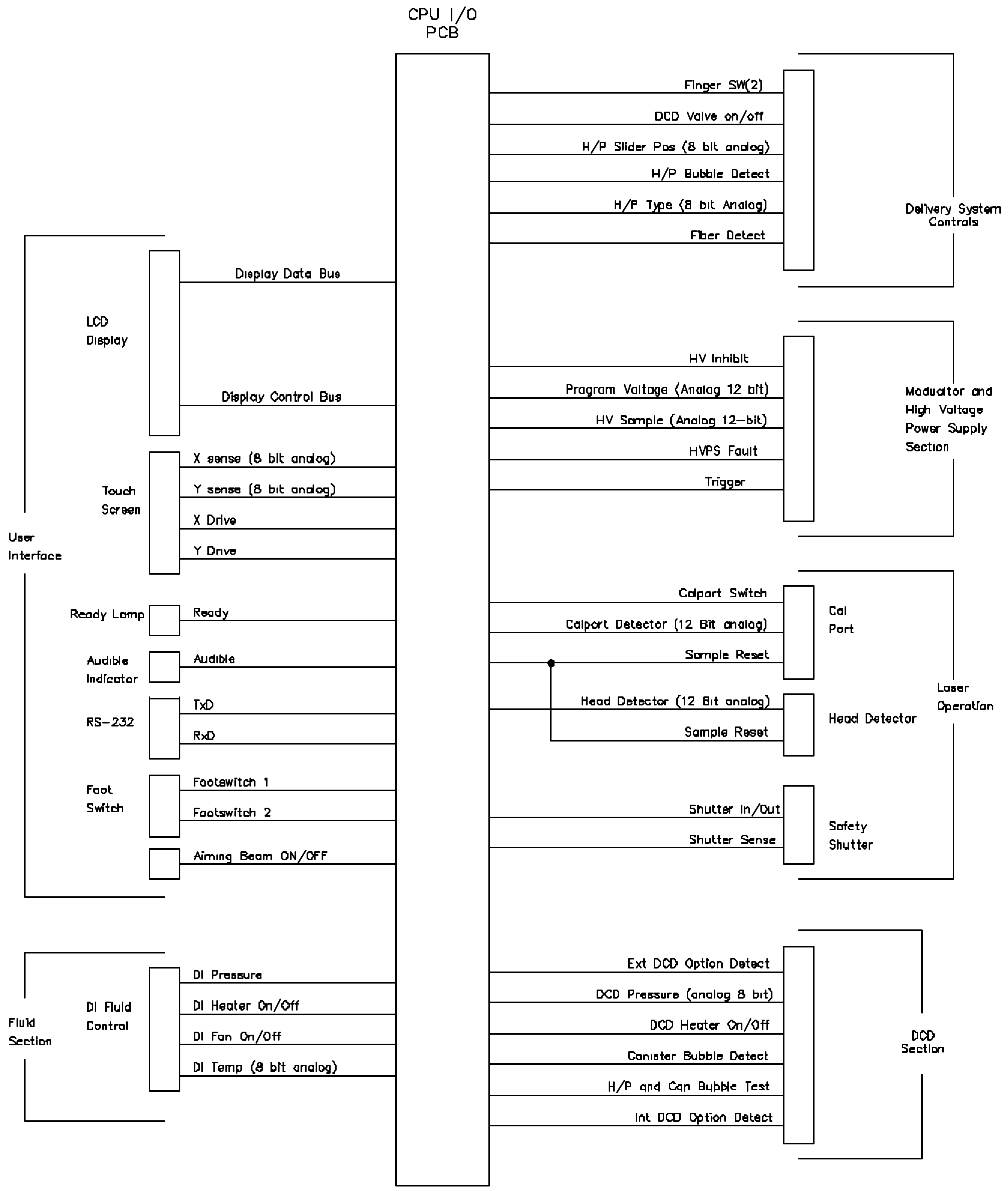
System Power Up:	4 short beeps, also serves as diagnostic test
Warm-up Complete:	1 short beep
Charging:	1 short beep
Pulsing:	1 short beep
Fault:	1 long beep

4.1.5. Footswitch

When the Footswitch is selected, the user must depress the footswitch to pulse the laser. If the footswitch is held down, the laser will continue to pulse at the selected repetition rate. The pneumatic footswitch is connected to two pressure switches on the CPU I/O board that are pressurized (activated) when the footswitch is depressed.

4.1.6. Fingerswitch

When the Fingerswitch is selected, the user must depress the fingerswitch on the handpiece to pulse the laser. If the fingerswitch is held down, the laser will continue to pulse at the selected repetition rate. The fingerswitch is actually two separate switches in the handpiece that are activated when the fingerswitch is depressed.



CPU System Control Diagram

5.0 CONTROL SYSTEM FIRMWARE

The control system firmware manages the transition of the code execution through the system states and processes. The diagram on the following pages show the MGL laser States and Processes.

5.1. Naming Conventions

- All non-volatile variables will be stored in a partitioned segment of battery-backed RAM that can only be erased by software. All variables stored in battery-backed RAM will be declared with the prefix "nv_" and will be defined in nv_ram.c. All other variables residing in RAM not declared as non-volatile will be cleared at power on.
- Graphics and icons have been used wherever possible to minimize language translation requirements with the firmware.
- All text or graphic bitmap images displayed on the UI will be located in files prefixed with "bmp_".
- All initialization functions called for system initialization purposes will begin with the prefix 'init_' or 'Init'
- The name of any interrupt level function will be terminated with the suffix "_int".
- The name of all *typedef enumerations* will be terminated with the suffix "_E"

5.2. Control System States

The MGL can be operated in any of three system states: *STANDBY*, *READY*, or *Fault*. The following are brief descriptions of the characteristics inherent to each state:

- STANDBY – when in the STANDBY state, lasing is inhibited.
- READY – when in the READY state, the system is enabled to lase and calibrate.
- FAULT – when non-typical system operation occurs, the fault handler inhibits lasing and displays a fault.

While transition between the *STANDBY* and *READY* states is controlled by both the user and the system, transition to the *Fault* state is controlled solely by the system.

5.2.1. State Changes

It is important that the system reverts to a known safe state whenever transitioning between states. To ensure this occurs, the following actions are executed when the system changes states:

- Shutter is closed.
- HVPS is disabled.
- High voltage reference voltage is set to 0.
- Charging interrupt is disabled and Charging state set to "Not Charged"
- Calibration Process reset.

The system will perform additional steps depending on the state it is entering.

1) Entering STANDBY
   - READY lamp is turned off.
   - State is set to STANDBY.
   - STANDBY timeout timer set
   - Calibration timeout timer set
   - Turn Aiming Beam Off.
2) Entering READY State
   - If the Footswitch is depressed, the system isn't allowed into Ready.
   - READY lamp is turned on.
   - READY state timeout counter initialized.
   - Rep rate timer initialized
   - State is set to READY
   - A two second delay is then implemented before proceeding as a safety precaution.
   - Turn Aiming Beam On
3) Entering Fault State
   - If a message is being displayed, the Main Menu is drawn.
   - READY lamp is turned off.
   - State is set to FAULT.
   - Turn Aiming Beam Off.

Mini-GL State Diagram

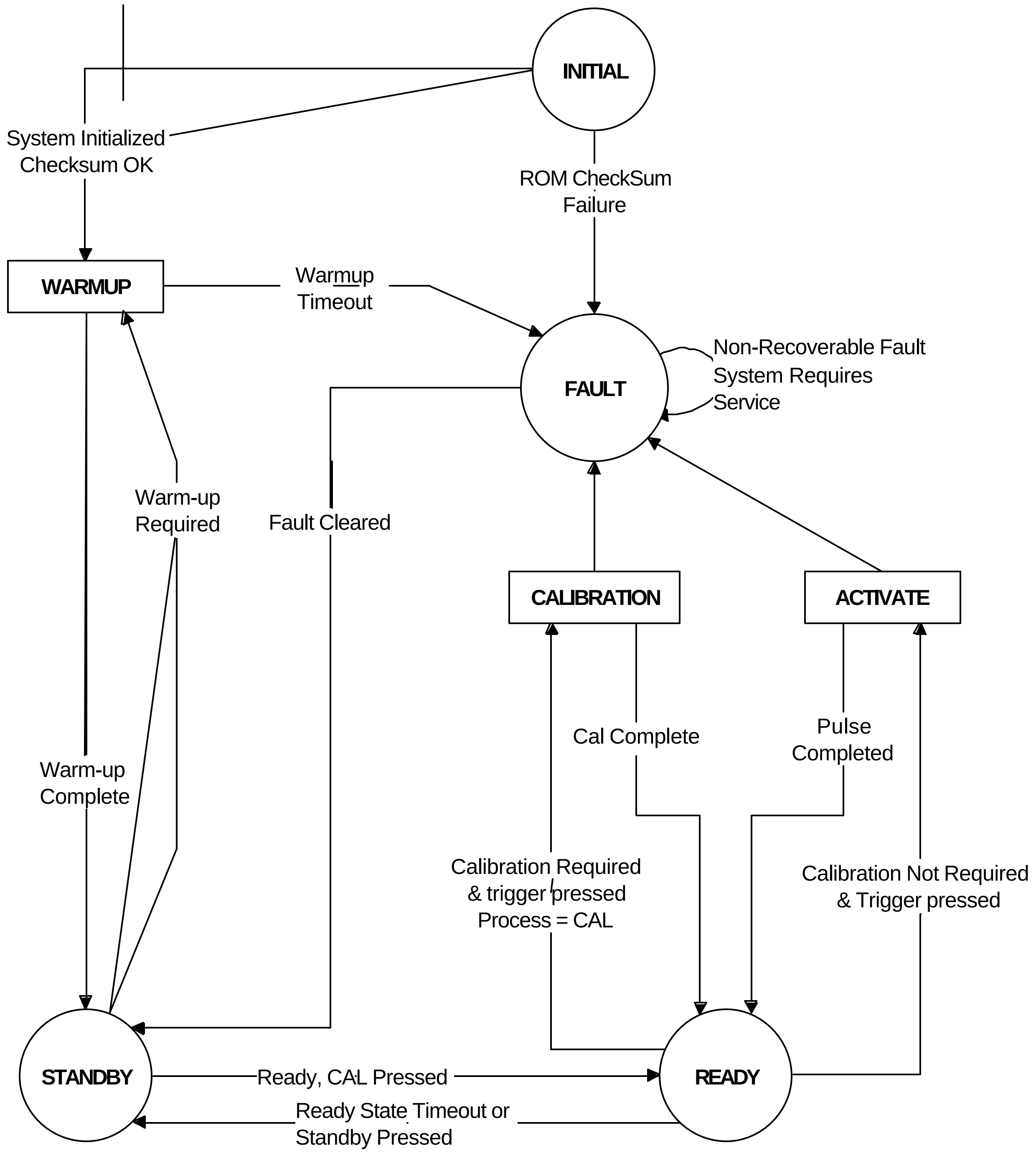


Figure 1: MGL Control State Diagram

5.3. Control System Processes and Procedures

The system states are general operation ideals to govern the action of the system. The system processes are specific to the individual states and perform a particular action.

The MGL Control System Firmware is comprised of the following processes and procedures. These are further described in Section 6.

1.	Init	-	Procedures initialize system hardware at power-up
2.	Warm-up	-	Controls heating of DI coolant and DCD cryogen as each approaches their respective operation setpoint and updates display with progress.
3.	Standby	-	Maintains system in STANDBY state.
4.	Ready	-	Maintains system in the READY state, and handles pulsing.
5.	Cal	-	Calibrates the laser to the fluence set by the user.
6.	Fault Handler	-	Processes and displays any faults or warnings that occur

Additionally the following supporting routines are used. These are described in Section 7 and throughout the text:

1.	Main Control Loop	-	Provides cyclical execution of system firmware.
2.	RTI Interrupt Routine	-	(Real Time Interrupt) System Timer.
3.	GUI Routines	-	Polls touch panel buttons, displays text and graphics.
4.	Digital & Analog I/O Routines	-	Translates digital and analog signals
5.	DCD Control Routine	-	Controls DCD cryogen pressure.
6.	DI Control Routine	-	Controls DI coolant temperature.
7.	Fault Check Routine	-	Checks for periodic faults.
8.	Calibration Routine	-	Calibrates laser to selected fluence.
9.	Pulse Routine	-	Pulses the laser once and dispenses a set amount of cryogen.
10.	Analysis Routine	-	Ensures that energy is within desired range.
11.	Maintenance Routine	-	Provides manual control of system hardware (service).

5.4. System Status Flags

There are 2 primary status flags used to monitor the status of the system. These flags are used by the processes to allow process synchronization. The State and Process flags are updated at the entrance to each process.

- STATE Flag indicates the system state.
  {STANDBY, READY, FAULT} only one active
- PROCESS Flag indicates the process or procedure that is executing.
  { WARM-UP, CAL, STANDBY, READY } only one active

5.5. Exception Handling: Fault Long-Jumps

The system may encounter operation conditions that are outside the system operation parameters. When these exceptions occur, the system must recognize and remedy the problem to return to specified operation conditions. The functions that monitor system parameters are normally subordinate to an associated control process. The monitoring function will call the *fault_handler* which will determine the proper actions to perform.

The software uses 2 C library routines known as *Setjmp* and *Longjmp*. These routines are used to simplify the exit from the *fault_handler* routine. The *Setjmp* function, executed immediately before the main control loop, saves the stack location and the address of the next command to be executed. After a fault is recognized and processed, the CPU performs a *Longjmp*. When the *Longjmp* occurs, the stack is reset to its state before the first iteration of the Main Loop and program execution resumes from the command following the *Setjmp*. In effect, the *Longjmp* recalls the stack and program execution parameters saved by the *Setjmp*. These functions eliminate both stack growth and the need to 'bubble up' through the routines that detected the fault.

5.6. Control System Firmware Architecture

5.6.1. Foreground Level

All foreground processes and routines will always execute in a sequential order. Background processes, however, execute at a regular interval independent of the foreground processes. All interrupt driven background process have priority to interrupt execution of the foreground processes of most routines.

5.6.2. Background Level

The background functions are prioritized amongst each other according to the Interrupt Vector Map on page 49 of the 68HC12 technical summary.

- Real Time Interrupt service routine – **rti_int**(), is periodic with a 1ms rate. It updates the software timer variables, reads the CPU's digital inputs, and maintains the watchdog timer.
- Serial port I/O service routine – **serial_int**(), supports the RS-232 communication between the laser firmware and the diagnostic maintenance mode software.
- Charge Timer routine – (Timer Interrupt Ch. 0), **Charge_int()**, while system is charging, the routine executes every 10ms. The routine monitors the voltage of the PFN to determine when charging has completed
- Output Timers used for output compare **duty_set()**. PortT of the HC12 allows Output Compare functionality in conjunction with timer synchronization for control of the associated pins. Although this is not an interrupt function, this control method is deemed 'background' because control of the state of the pins is not performed by firmware rather by CPU timers.

6.0 CONTROL SYSTEM PROCESS DESCRIPTIONS

In depth description of each process and procedure within the control system firmware are presented in this section.

6.1. Initialization Procedure

The initialization procedure amounts to the group of functions executed preceding the Main loop. The goal of the procedure is to call all routines necessary to initialize each of the laser sub-systems and HC12 operation parameters.

1) Initialize the memory expansion mode of the HC12 and Real Time Interrupt to the desired interrupt rate.
2) Initialize the function of all pins as either Input or Output and set all the hardware (outputs) to a known state.
3) Initialize the CPU on-chip peripherals (SPI, SCI, ADC)
4) Initialize the LCD controller and displays logo, software part number, and the revision number.
5) The firmware performs a ROM checksum test each time the laser is powered up. The checksum for the ROM is calculated and compared against the checksum value stored in the ROM. If the checksum test fails, the appropriate fault code is displayed and the firmware cycles in an endless loop.
6) Test non-volatile system variables for corruption.
7) Test both the audible alarm (4 short beeps) and READY lamp.
8) Initialize global parameters.
9) Initialize and reset both the HP limit table and the Fluence Limit table. Determine the handpiece connected to the lasers.
10) Reset both the DCD and DI PID data structures, initialize output compare on channel 7, and test both the DI pump and the DI pressure switch.
11) Enter the Warm-Up sub-process.

6.2. STANDBY Process

The STANDBY process is simply a dummy process to be entered while system is inactive in the STANDBY state, i.e. not in Warm-up process. The process maintains a few flags and ensures the system remains inactive and in a 'safe' state. The following actions are taken while in the STANDBY process:

- If the calibration timer timeouts, a cal required flag is set.
- HVPS is disabled and High voltage reference voltage is set to 0.

Ensures the Main screen is displayed

6.3. Warm-up process

The Warm-up Process executes a small state machine cycling through the states shown below in numerical order. The process simply holds the system in a state where user input is disabled while displaying the status of the DCD and DI systems as each approaches its respective operating pressure and temperature. The warm-up process is entered after system initialization, if a warm up is required on exiting MM, and when a fault is cleared by the user and a warm-up is required. The following are the Warm-up states:

1) BEGIN
    - The warm-up message box which contains a bar graph is displayed.
    - Automatic control of both the DI and DCD systems are enabled.
    - Initial percent completion value for the warm-up cycle is calculated and displayed.
    - The system time where warm-up begins is buffered for use in Warming sub-process.

2) WARMING
   - Software calculates the progress of the warm-up based on the system (DI or DCD).
   - The Warm-up Status bar graph is updated based on whichever parameter is further from operations range.
   - This sub-process monitors the time required to reach the respective operation ranges of the DCD and DI systems. The sub-process will trigger a fault if the warm-up time exceeds 60 minutes.
3) AT TEMP
   - The warm-up process exits solely on the status of the DI temperature when DCD is not enabled. If the DCD is enabled, the DCD pressure and the DI temperature must both attain their warm-up set points.
   - Warm-up required flag is reset.
   - Warm-up state machine is set to DONE
   - System control is returned to the stand-by process of Stand-by state
   - Warm-up screen is cleared and the Main screen is displayed.

6.4. READY Process

The READY Process is entered when the system is inactive in the READY state. The following procedures are performed when the system is in the READY process:

- The READY State timeout is checked to determine if the laser has not been used in the last two minutes. This is a safety measure to prevent accidental use of the device.
- The process continuously monitors the status of the footswitch to determine when to call the Activate routine or Cal processes.
- Process is responsible for initiating the charge process when PFN is Not Charged.

6.5. Activate Procedure

The Activate procedure coordinates the verification of the footswitch status, DCD spray, pulsing the laser, and analyzing data. The procedure can be entered from both the Cal and READY processes.

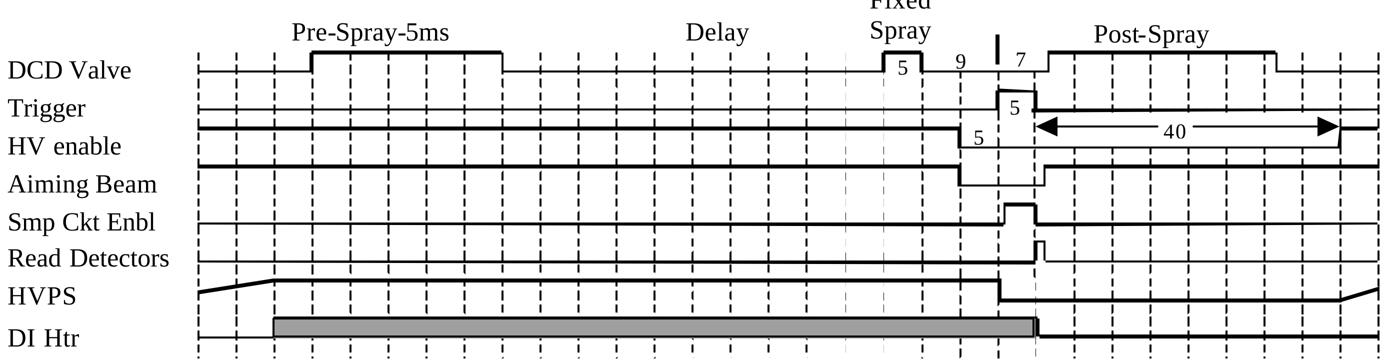
The procedure is passed a status flag informing the process if the laser is already enabled (i.e. in a multi-pulse sequence). The shutter is verified on each pulse. The laser is pulsed after the system performs the DCD prespray, and DCD delay. The postspray is completed, and then calls the *pulse_analysis()* routine, updating the READY state timer and returning to the calling process.

Pulse Timing

The pulse routine, **fire_pulse**(), is called from the Activate process or from CAL and performs the following tasks:

- Blanks the Aiming Beam (prevents possible problems with laser light feeding back into Aiming Beam's active power control).
- Disables the HV power supply.
- Triggers the laser.
- Samples the energy circuits after a delay.
- Resets the Rep timer to the appropriate time. (Dependant on DCD spray and delay timer)
- Enables Aiming Beam
- Updates system and head pulse counter.

DCD Spray
PreSpray – Shown as 30ms
Delay – Shown as 50ms
PostSpray – Shown as 30ms
grid = 5ms



DCD Spray during HV Charge

There are three functions that support the ability of the laser to start the DCD pre-spray while still charging. This allows the laser to pulse repetitively based on the HVPS charge time, and be less impacted by long DCD pre-spray and delays. Since the HV charge time remains the same for a set voltage, it records the last HV charge time, and uses that time to determine the beginning of the next pulse's pre-spray segment. A HV charging estimate is used the first time, so the same algorithm can be used.

- EstimateChargeTime() – Estimates the HV charge time based on a linear slope of 0-2000V = 0-1.1sec using the ratio of ChargeTime/Max Charge Time = Charge_voltage/MAX_HVPS. Adds in a safety factor, and recalculates the repetition rate time
- GetChargeTime() – Get the charge time of the last HV charge (prior to first charge, would return the estimate).
- ResetRepTime() – Sets the repetition rate based on the charge time and pre-spray/delay time.

Pulse Analysis

After a successful pulse, the data acquired during the pulse is compiled and analyzed. This data is used to determine if the laser is operating within system specifications. The following occurs when a pulse is analyzed:

- HD and CP energies are calculated from the applicable detectors.
- If the process is not calibration, a HD energy check ensures energy is within ±10% of the calculated xHD.
- If the hand piece is detected in the cal port, the transmission is checked.

6.6. Calibrate Process

The calibration process is initiated when:

- System is in STANDBY or READY and the CALIBRATION command button is pressed.
- System in READY, a calibration is required, and the active trigger switch has been depressed.

Calibration Overview

The calibration process determines the charge voltage necessary to generate a pulse with the energy that matches the front panel setting. To accomplish this the system must:

- Determine the relationship between energy vs. charge voltage for the installed delivery system. (Slope Calculation)
- Calculate the desired delivered energy based on the Fluence set on front panel and HP spot size.
- Calculate an initial charge voltage based on the desired delivered energy and the Energy vs. Charge voltage slope. After obtaining pulse data, the system must change the charge voltage based on the difference of the measured energy from the desired energy.
- Pulse until the CP energy is within a percentage of the desired energy.

During this process the system instructs the user on the correct action to perform and checks the energy data to ensure the system is operating within specified parameters.

Calibration Required Conditions

The following is a list of conditions that will require a calibration when attempting to pulse after entering the READY state.

1. System Power-Up: On all system power-ups, an initial calibration is required.
2. Pressing the Calibrate Button: The calibrate button automatically puts the laser in the READY state and sets the calibration required flag before entering the calibration process.
3. Changing Fluence: Selecting the fluence parameter and pressing either the up or down buttons causes the calibration required flag to be set.
4. Exiting Maintenance: A calibration is required when exiting from MM.
5. Hand Piece Connector: When the detected HP is disconnected or in an error state, a calibration will be required when a valid HP is connected to the system.
6. Calibration Timeout: Parameters governing system performance can change over time. The calibration timer is set when the system enters STANDBY. If the timer expires while still in STANDBY, a calibration is required.
7. Fault Conditions: A number of faults only occur when the output energy characteristics of a system change. These changes require the system to re-calibrate. The following is a list of faults requiring a recalibration.
    - HV Tolerance Fault
    - Calibration Fault
    - DI Temperature Fault
    - Delivery System Fault
    - Energy out of Range (EOR) Fault
    - Transmission Fault

Calibration Initialization

The system instructs the user to perform a number of actions when entering the calibration process to ensure system inputs are functioning correctly and the system is safe to pulse. The following instructions are displayed on the display when the calibration process is initiated:

- "CONFIRM SPOT-SIZE IS xxmm" displays to ensure user checks has correct spot size (not shown on MGL-LE)
- "INSERT HANDPIECE INTO CAL PORT" appears if the hand piece is not inserted in the CAL PORT
- "DEPRESS FOOTSWITCH" appears when the hand piece has been inserted into the CAL PORT.
- "CALIBRATING" appears once the trigger has been activated.

Fingerswitch Trigger

Since the fingerswitch cannot physically be pressed with the handpiece in the cal port, an alternate button is displayed on the TSDP. This button must be pressed and held for the entire calibration.

Slope Calculation

A slope calculation is the first half of a 'Full Calibration'. Its purpose is to determine the relationship between the CP Energy and (Charge Voltage)^2 by measuring the CP energy for two distinct charge voltages. This slope differs for each of the spot sizes. The slope is later used to make adjustments to the charge voltage to achieve the desired energy.

Calibration Process

The calibration process is quite simple. The system calculates an initial charge voltage to begin with using the desired energy and the E vs V^2 slope. The system pulses and analyzes the data obtained from the pulse. A new charge voltage is calculated based on the difference between the desired energy and the energy measured during the preceding pulse. The system is charged to the new voltage and process is repeated until the delivered energy is in range.

Charge voltage changes are made based on the desired energy, the measured energy, and the charge voltage of the preceding pulse.

$$\Delta E = E_{Desired} - E_{LastMeasured}$$

$$New\_V = \sqrt{V_{Last}^2 + \left(Slope * \Delta E\right)}$$

Equation 1 - New Voltage Calculation

Throughout the calibration process the system continuously monitors the CP/HD transmission to ensure the system does not damage the delivery system.

If the CP energy is within ±5.0% of the desired energy, the calibration is successful. The system calculates an expected Head energy (xHD). All treatment pulses are measured against this value to ensure the energy delivered by the laser stays within a ±14% range of the head detector. This energy is the basis of the Energy out of Range (EOR) fault. The xHD value is calculated using the expected CAL Port (xCP) energy and the means of the HD and CP energies.

$$XHD = (xCP/CP) * HD$$

Equation 2 – Calibration Head Lock-in value

Calibration Complete

The following instructions are displayed on the display when the calibration process is completed:

- "RELEASE TRIGGER SWITCH" instructs to release trigger switch
- "REMOVE HANDPIECE FROM CALPORT" instructs the user to remove handpiece – this ensures the Cal Port switches are working correctly (no DCD Spray if handpiece is still detected in Cal Port). The system will revert back to Standby to prevent inadvertent trigger press while installing distance gauge.

Calibration Warnings

There are a few calibration warnings that may be encountered during a calibration:

- "MAX FLUENCE" appears when the laser's HD energy would be too high to attain the selected fluence. The fluence is lowered to a HD energy that is achievable. The new lower fluence is displayed with the message. The TSDP fluence settings will not permit any fluences above this fluence until the power is cycled, or a full CAL is initiated.
- "EXIT TO CLEAN WINDOW?" appears with Yes/No buttons when the transmission is <75%. The YES button allows the user to clean the handpiece window, and try recalibrating, while pressing NO will continue in the present CAL.

6.7. Fault Process (Includes Warnings)

The Fault Process handles all fault and warning conditions. Warnings are considered malfunctions that can be easily corrected by the operator. Faults indicate system operating outside of acceptable parameters due to system misuse or malfunction. Persistent faulting will likely require a service call to correct the problem.

Faults are displayed with separate message windows displaying the numeric fault code and an OK command button. If more than one fault exists, the firmware will continuously cycle through each numeric fault code at every 3 seconds. Depressing the OK command button will clear the fault(s).

Faults are registered when the system is not in maintenance mode and the fault is not active. When in maintenance mode, the **ignore faults flag** disables fault handling.. In this case no action is taken by the fault handler and program control is returned to the point when the fault was detected.

There are three tables that are used in the fault process to log fault occurrences, determine processed faults, and determine which process must occur after a fault is cleared.

- The *Fault Attributes* table contains attributes associated with that fault or warning indicating the required system actions for particular faults. The attributes are warm-up required and calibration required.
- The *Fault Status* table contains flags indicating a fault has been processed. The cyclic operation of the main control loop results in repeated calls to the fault handler for certain faults. Once a fault has been processed the fault handler will ignore subsequent occurrences of the same fault until that fault is cleared.
- The *Fault Log* contains a event counter for each fault. When a fault is encountered, the corresponding counter is incremented. The counter stops incrementing when 255 event (faults) have been logged. The Fault Log is stored in non-volatile RAM. The data can be viewed in Maintenance Mode.

6.7.1. Fault Checking

The check_faults() routine periodically checks for a number of fault conditions. The remaining faults are checked by the code of each system sub-module. The point at which each fault is checked in the program execution is given in Warning and Fault Descriptions below. When a fault is detected in the firmware, the Fault Handler is called.

6.7.2. Fault Handler Routine

Faults are only processed when the system is not in maintenance mode. When a fault that has not been processed occurs, the fault_handler() routine generates a beep tone to alert the user. The system then sets the fault status flag in the fault_status[ ] table to indicate that the fault has been acknowledged/processed. Next the fault count in the nv_fault_log[ ] table is incremented to record the occurrence of the parent fault (max. Count = 255 per fault). The fault attributes in the fault_attributes[] table are checked and the corresponding system actions are taken (i.e. turn off DCD heater, set cal_required flag, etc.). The *state_change* routine is then called to return the system to a safe non-lasing state and enter the Fault State. After the fault message box displayed is drawn, code execution is returned to the beginning of the main process loop using a longjump operation.

Once a fault has been processed, the fault handler will ignore subsequent occurrences of the same fault by returning program control to the calling routine for faults that have already been processed. The REPLACE CANISTER and the PURGE REQUIRED warning messages are treated as special cases. None of the fault tables are used. The beep tone is generated and the unique message boxes are displayed. Program control is returned to the beginning of the main process loop using a longjump operation.

6.7.3. Interrupt and Background Function Cleanup

Several maskable interrupts and background functions are used throughout the software for detection and monitoring of tasks that would hold up normal system operation. Some interrupts are required to execute under all conditions. Those background functions and interrupts which are not required while an interrupt is active must be disabled. Given a fault occurs before a background function or interrupt has been completed, it is the first priority of the fault handler to disable the interrupts and any processes or I/O controlled by the interrupt.

6.7.4. Fault Numbering

A number of the MGL faults have several potential causes. To aid in the analysis of faults and system malfunctions, sub-fault numbering exists for all faults with multiple causes. A general fault number addresses the sub-system or process where the fault occurs. The sub-fault addresses the particular cause of the fault. All faults will be categorized and displayed to the user in this format: *Parent_Fault #* **.** *Sub-Fault#* e.g.10.2. Given a fault does not have any sub-fault, the fault sub-fault number will = 0.

7.0 SUPPORTING ROUTINES

7.1. Main Control Loop Routine

The *Main Loop* continually executes the code of all major sub-systems. All foreground routines are either embedded in the *Main Loop* or subordinate to a routine that is. After all sub-systems are initialized, program control enters the *Main Loop* where periodic routines, GUI routines, communications routines, and state control routines are called in sequential fashion. When the system encounters a system fault, program execution is set to resume from the beginning of the *Main Loop*.

7.2. Digital & Analog I/O Routines

Although the 68HC12 enables direct reads and writes to/from the ports that control the inputs and outputs of the laser system, it is difficult to remember what I/O is assigned to which bit of what port Within system firmware there exists a soft layer to serve the purpose of:

- Keeping track of the state of each input and output.
- Filtering out signal noise when reading inputs.
- Allowing Inputs and Outputs to be referenced by names associated with the function.

This soft layer is also used to control analog inputs and outputs of the system:

- A single channel 12-bit D/A, to set analog voltages.
- A four channel 12-bit A/D for measurement of signals requiring high resolution.
- The 8 A/D channels of PORTAD of the CPU which provide additional low resolution(8bit) analog measurements.

The 12-bit A/D and the 12-bit D/A devices interface with the CPU via a Serial Peripheral Interface (SPI) bus. The following firmware routines are used to read, write, measure, and set the various I/O and analog signals associated with the system:

7.2.1. Digital Input Routine

The digital input routine, get_input(), enables the firmware to detect the current logic state of each external input to the CPU. The routine ‘debounces’ each input to its active logic state by performing a logical AND of the three most recent port samples stored in the input buffer. This buffer, updated every 10 ms within the Real Time Interrupt (RTI), buffers the values of all port registers with pins configured as inputs. All active-low inputs are inverted prior to the AND operation. Firmware then performs a logical AND of the processed port image and the mask of the pin of interest to return the state of the input. The routine also update the *input_status* word used by Maintenance Mode.

7.2.2. Digital Output Routine

The digital output routine, set_out(), allows the CPU to control the state of individual hardware outputs. The routine is designed to be passed the desired active state(Active, Inactive; On, Off; Enabled, Disable) of the output rather than the logic level. This reduces the amount of logic, external to the routine, required to correctly control the output as well as removing the burden from the programmer of remembering the active state of each output. The routine also updates the *output_status* word used by Maintenance Mode to indicate the state of system outputs.

7.2.3. 8-bit Analog Input Routine

The 8-bit analog input routine, convert_8bit(), reads the value of an individual input channel to the CPU’s analog converter (PortAD). The channel to be read is passed to the routine as an input parameter. The routine writes a conversion command to the ATDCTL5 register to start the conversion. It then polls a ‘conversion complete’ flag bit in the ADTSTAT register. When the flag is set, the conversion data is read from the data register. The routine returns the 8-bit data as an *unsigned character* data type to the calling routine.

7.2.4. 12-bit Analog-Digital Acquire Routine

The 12-bit A/D is used to acquire inputs from the energy detectors and HVPS. The 12-bit analog input routine, convert_12bit(), reads the value of individual inputs of the 12-bit A/D. The desired channel is passed to the routine as an input parameter. The routine writes a conversion command byte to the ADC with the type of conversion to be performed and the channel to read. The transmit “done” byte is polled after each byte write. When the flag is set, the SP0DR register is read. The first byte received from the ADC is ignored. To receive data, the system must send out a dummy byte with all zeros. The second byte received contains the upper 7 bits of conversion data. The lower 5 conversion bits are returned in the second receive byte after the second dummy byte. The data are then shifted to the left and right, respectively, and combined into a unsigned integer data type. The routine returns the 12-bit result to the calling routine as an unsigned integer.

7.2.5. 12-bit Digital-Analog Output Routine

The 12-bit analog output routine, set_dac(), sets the digital to analog converter (DAC) to produce an output voltage. For the MGL laser system, the output of the 12-bit DAC is used to set the HVPS reference voltage which determines the PFN charge voltage. The desired HVPS voltage is entered into a transfer function which determines a 12-bit value that is passed to the set_dac() routine. Once the routine is entered, the firmware sets the DAC chip select line low. It then writes the upper 8-bit word to the SP0DR register. The transmit 'complete' byte of the SPI status register(SP0SR) is polled after each byte write. When the first transmission is complete, the lower word is written. When the second transmission is completed, the SP0DR register is read and cleared. The Chip select line is set high and the output of the DAC assumes the new value. The routine returns to the calling function.

7.3. Interrupt Routines

The firmware utilizes the CPU's non-maskable COP watchdog reset as well as maskable interrupts: Real Time Interrupt, SCI 1 (Serial Communication Interface) and the interrupt on timer channel 0. Of the maskable interrupts, highest priority is given to the Real Time Interrupt, all other interrupts follow in priorities assigned in the Interrupt Vector Map, Table 17 of the MC68HC12 technical summary p.49.

7.3.1. Real Time Interrupt (RTI)

The Real Time Interrupt service routine, rti_int(), is a periodic interrupt which executes once every millisecond. It updates the software timer variables, reads the CPU's digital inputs, and maintains the watchdog timer. Timer variables can be designated with 1ms, 10ms and 1s update rates. Depending on the application, the timer may be incremented or decremented by the routine.

A LIFO, last in first out, input buffer (table), input_buf[], is maintained by the routine. It is updated every 10ms with the current state of the hardware inputs (digital) from the Laser I/O PCB. The input buffer is used by the get_input() routine to determine the state of an individual input.

The COP Watchdog Timer is also maintained by this routine. The routine decrements the Foreground Timer once a millisecond. The Foreground Timer is reinitialized at various points in the firmware during the execution of the main control loop. Provided that the Foreground Timer has not expired (reached zero), the COP Watchdog Timer is reset every 10ms by the routine and normal execution continues. In the event that the main control loop running in the foreground should hang or stop operating correctly, the Real Time interrupt routine would allow the COP Watchdog Timer to expire. This would result in a non-maskable CPU reset.

7.3.2. Charge Interrupt

The firmware charges the HV to a set voltage while minimizing the length of the charge process. While charging, the firmware constantly monitors the voltage of the PFN to determine the End of Charge (EOC), and also that the HV meets any tolerance requirements.

The firmware periodically monitors the charge voltage of the PFN as a background process, thereby allowing foreground process execution to continue. Allowing continuous program flow through the system *Main Loop* while charging preserves the schedule of periodic routines; control and monitoring of system I/O; as well as maintenance of external communications and touch screen interface.

The background process uses the standard timer module of the 68HC12 and the ability to link the timers with hardware interrupts, allows execution of a periodic routine asynchronous to the *Main Loop* execution.

The charging interrupt is enabled by the InitCharge() routine. After checking all of the conditions required for the charge process to take place, the initialization routine sets the HV reference, sets the PFN status to CHARGING, and enables both the interrupt and associated timer channel.

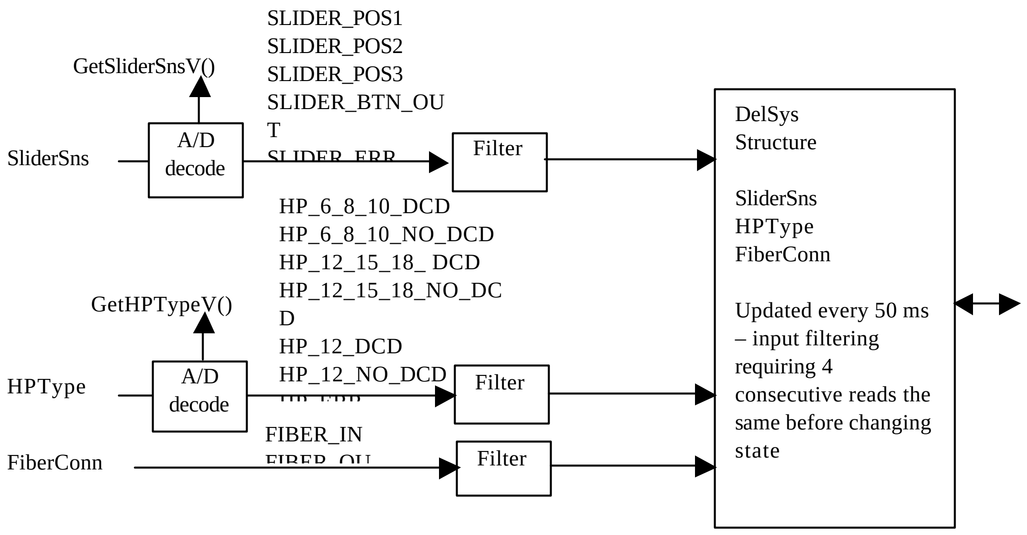
Once the interrupt is enabled, a call to the Charge_int() function is made when the free-running timer TCNT equals the value of the timer register TC_x. Each time the routine is entered it:

- Resets the interrupt flag.
- Resets the time for the next interrupt to occur.
- Increases the routine iteration count.

The system acquires the current PFN voltage (HVsmp) and enters it into a LIFO buffer. When the last three samples all fall within 20V of each other, the PFN voltage is determined to have reached EOC. The last 3 samples are then averaged and checked if are within a percentage of the HVref. If the HVsmp is within the charge window, the charge is complete. The interrupt is disabled and the PFN status is set to CHARGED. The interrupt is also disabled when the charging process has failed due to the HVsmp never reaching an EOC, or the HVsmp is not in tolerance.

7.4. Delivery System

The Delivery System module (hp.c) contains functions that manage the spot size sense, fiber connected, aiming beam intensity, and fluence limits. The CheckDeliverySys() function, called periodically, updates the delivery system parameters and checks for any faults.



The following functions control the Fluence limits. Each handpiece has different fluence range and resolutions; also the upper fluence limit can be changed during a CAL if it encounters a “Max Fluence” condition.

- InitHPFluenceLimits() – Sets default handpiece fluence limits
- GetHPFluenceLimit() – Gets the fluence limit (low or high) for the specified spot size
- SetHPFluenceLimit() – Sets a new fluence limit (low or high) for the specified spot size (Max Fluence uses)
- ChangeHPFluence() – Changes the fluence (up or down) with consideration of the fluence resolution and limits
- IsFluenceSettingValid() – Tests if specified fluence setting is in range and has correct resolution for this spot size

These functions convert their associated analog signals to voltage values. The values are displayed in MM:

- GetHPTypeV() – Gets Handpiece Type voltage
- GetSliderSnsV() – Gets Slider Sense voltage

These functions process the delivery system inputs:

- GetHPSliderSns() – Converts and filters the analog signal to slider position 1-3, button out, disconnected, or error
- GetHPType() – Converts and filters the analog signal to 6-8-10mm, 12-15-18mm, 12mm (with DCD or no DCD) handpieces, disconnected, or error (MGL-LE only supports a 12mm spot size).
- GetFiberConnected() – Reads and filters the fiber switch position

These functions provide control of the Aiming Beam. The aiming beam isn’t physically part of the delivery system. However, since the intensity is directly related to the handpiece spot size, it was placed with the delivery system functions:

- ChangeAimIntensity() – Change the aiming beam intensity based on the spot size
- GetAimIntensity() – Gets the aiming beam PWM duty cycle for the specified spot size

These functions provide information on the spot size that is being decoded:

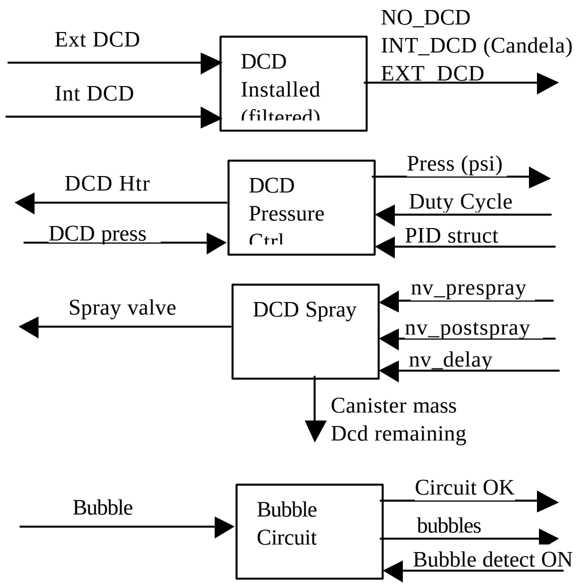
- GetHPSpotIndex() – Gets the current spot size selected, or the last spot size selected if button out or error
- IsHPSpotValid() – Tests if the spot size is valid
- GetHPSpotArea() – Gets the selected spot size area (cm2)

This function manages the DCD related components of the handpiece. All DCD spray control is managed in the DCD module.

- IsHPWithDCD() – Tests if the handpiece supports DCD or not

7.5. DCD Control Routines

The Dynamic Cooling Device sprays a coolant (cryogen), prior to, or after the lasing pulse to reduce damage to the epidermis and possibly reduce pain. The coolant is supplied from a canister, maintaining the canister at a temperature above ambient. The higher temperature is necessary to prevent the cryogen from gassing when passing through the handpiece. This system maintains the temperature using a heater, and regulating the pressure. This module also supports multiple bubble sensors, spray valve actuation, and canister cryogen level information. See Appendix D for a diagram depicting the DCD controls.



7.5.1. Installation Types

The system supports three different DCD installation types. The system power should be cycled after changing an installation type.

- NO_DCD – If the DCD connector is not connected, the laser won't allow DCD spray. The DCD related Buttons (Purge, Pre-Spray, Delay, Post-Spray, Canister Volume) are not displayed, and a spot-size icon is displayed in place of the Purge button. The firmware will set the DCD Canister Heater signal Off (safe condition.)
- INT_DCD – If the internal DCD is connected (primary DCD control signal), the cryogen canister pressure will be maintained, and DCD spray is allowed. The DCD related Buttons (Purge, Pre-Spray, Delay, Post-Spray, Canister Volume) are displayed and active. If a DCD enabled delivery system is not connected to the system, don't display buttons.
- EXT_DCD – If an external DCD is connected (secondary DCD control signal), the cryogen canister pressure is not maintained, but DCD spray is still allowed. There is no canister, canister heater, temperature sensor, or bubble sensor present, The cryogen is supplied from an external source. The DCD related Buttons (Purge, Pre-Spray, Delay, Post-Spray, Canister Volume) are displayed and active. If a DCD enabled delivery system is not connected to the system, don't display buttons.

7.5.2. Pressure Measurement

Pressure is measured each time the *DCD Pressure Control* routine is entered. This routine reads the value of the A/D channel associated with the DCD pressure transducer. The raw A/D value is processed to result in an instantaneous pressure reading. This routine maintains a buffer of the 4 most recent pressure measurements. To decrease the system response to noise, the samples are averaged and returns the processed value.

7.5.3. Pressure Control

The measured pressure is input into a PID routine that generates a control signal (duty cycle) to modulate the pulse width of the output signal. The canister heater temperature is controlled, thus maintaining the DCD pressure. Cooling the canister (lowering pressure) can only be achieved by convection cooling.

The system is set to operate at 120 psi, with an acceptable operating range from 105 psi to 135 psi. If the pressure fails to stay within the acceptable range after exiting the Warm-up process, a DCD over/under pressure fault is flagged by the firmware. When the DCD pressure is out of range, the system overrides the generated duty cycle and disables the canister heating until the fault conditions no longer exist or the system enters the warm-up process.

DCD Pressure Range	Heater State	System Status
$0 \le$ Press < 25 PSI	0.0%	No canister installed
$25 \le$ Press < 105 PSI	100%	Canister installed – warming
$105 \le$ Press < 135 PSI	Automatic	Normal canister range
$105 \le$ Press AND in READY	Automatic	Fault - Under pressure
Press $\ge$ 135PSI	0.0 %	Fault - Over pressure

Table 1: DCD Pressure Control

7.5.4. Spray

When enabled, a cryogen spray is emitted from the handpiece nozzle with each laser pulse. The spray length and sequence for each laser pulse is designed to be the same as GL+, and consists of the following:

- Pre-Spray time [ms] – 5ms (from TSDP)
- Delay time [ms] (from TSDP)
- Fixed Spray time [5ms]
- Fixed Delay time [9ms]
- Laser Pulse
- Fixed Delay time [~7ms]
- Post-Spray time [ms] (from TSDP)

Cryogen can be manually sprayed using a purge button. The spray time is the total of Pre-spray + Post-spray time. The bubble sensors are ignored during this purge.

7.5.5. Bubble Sensors

There are two bubble sensors, with one located at the canister outlet, and the other inside the handpiece. The sensors ensure that bubbles are not in the cryogen line when DCD is being sprayed. While the valve is open and cryogen is spraying, it continuously monitors the bubble sensors (canister and/or handpiece). The percentage of bubbles in the cryogen spray is calculated as a ratio of the number of reads with bubbles detected to total reads for each spray segment. A fault is generated if the bubble percentage exceeds a specified threshold (>15%.) The canister sensor is not monitored when the EXT_DCD installation is selected.

A bubble fault can occur with the following conditions:

- If the handpiece bubble sensor detects bubbles while spraying during a pulse, a “Purge” message is displayed, and the system goes to Standby. The “Purge” message remains until the either the there are no bubbles when the purge button is pressed, or the cancel button is pressed.
- If the canister bubble sensor detects bubbles while spraying, the “Replace Canister” message is display, and the system goes to Standby. This message is also displayed the first time the Canister Pulse Count = 0.

The bubble sensor circuit is tested prior to use to prevent lasing pulses without requested DCD spray. The circuit simulates bubbles, and checks that the bubble sensors both detect them. This test is done on entry to READY, and also when the total spray time increases from 0.

7.5.6. Pulse Count Calculation

The DCD pulse count provides the user with the approximate amount of remaining cryogen in a canister. The pulse count is calculated using the initial amount of cryogen in a canister and the amount of cryogen displaced when sprayed.

The MGL canisters are filled with a minimum nominal amount of 1000g of cryogen. Due to the design of the canister the entire cryogen supply is not available to spray. There is a minimum canister volume (or mass) that an acceptable amount of cryogen spray is available. This remaining mass, referred to as the *Cryogen Reserve Mass* is determined using this formula:

$$Available\_Cryogen_{FullCan} = Cryogen\_Mass_{FullCan} - Cryogen\_Mass_{RESERVE}$$

Equation 3 - Available Cryogen in Full DCD Canister

The full can *Available Cryogen* mass is used as the initial mass of new canister. The user has the ability to select total DCD spray times from 10 – 200 ms in 10ms increments. Longer spray times result in more cryogen being displaced from the canister. It was determined that the DCD deposition rate (mg/ms) remained linear over the entire spray time range.

$$DCD\_Displacement(mg) = DCD\_Deposition\_Rate(mg/ms) * DCD\_SPRAY(ms)$$

Equation 4 - DCD Displaced per Pulse

For each DCD spray, the *DCD_Displacement* mass is subtracted from the *Available_Cryogen* mass. The DCD remaining pulse count is then calculated from these two values.

$$DCD\_Pulse\_Count = \frac{Available\_Cryogen(g)}{DCD\_Displacement(mg)}$$

Equation 5 - DCD Pulse Count Calculation

7.6. DI Control Routines

The DI control routine maintains the temperature to the DI setpoint of 65°C. The DI control system consists of a thermistor, heater, and Heat Exchanger Fan.

DI Temperature Range	Heater State	Fan State	System Status
0 < 5°C	OFF	OFF	DI Thermistor Shorted fault
5°C ≤ Temp < 65°C	ON	OFF	
65°C ≤ Temp < 98°C	OFF	PWM	
Temp > 98°C	OFF	OFF	DI Thermistor Open fault

Table 2: DI Temperature Control

7.6.1. Temperature Measurement

The DI temperature is measured in the same manner as the DCD pressure. The system simply reads a different A/D channel and uses a different transform to convert from the A/D value to an actual temperature.

7.6.2. Heater Control

The heater control is a simple 'Bang-Bang' controller that provides sufficient control to heat and maintain the DI coolant to the set point temperature. This method provides the fastest and most reliable heating time, while minimizing the overshoot.

The heater temperature control is overridden for the following conditions:

- The heater is switched off to limit the power from the HVPS-24VDC when the system is in the READY state and the laser is not charged or the trigger switch is depressed.
- The heater is switched off to prevent an overheating condition of the fluid system when the DI pump is in a faulted state.

7.6.3. Heat Exchanger Fan Control

A fan is used to regulate the DI temperature when it is above the set point. PWM is used to control the fan speed, and subsequently the rate of cooling, to allow a more precise and stable control of the DI temperature. Employing PWM with a fan also results in quieter system operation. The HX fan duty cycle is generated by a PID routine using the DI temperature measurement as an input.

The fan is also cycled on briefly (3 seconds) after a prolonged period in Standby (3 minutes). This is intended to prevent the DI compartment from building up heat and potentially causing the temperature of the DCD fluid lines (in the compartment above) to be higher than the canister temperature and causing bubbles.

7.7. PID Calculation

The PID routine is used by both the DCD and DI control routines for control of the duty cycles of DCD heater and HX Fan, respectively. Associated with each of these controls is a PID data structure which contains all pertinent information to be able to implement PID control: P,I,&D control gains, control variable set points, biases, integrated error, current control signal, etc….

The following is the procedure followed for all PID calculations. Each PID process is passed a different control data structure.

Algorithm Implementation

The control signal is calculated using the error of the variable of interest from the control variable setpoint.

$$error = SETPOINT - Control\_Variable$$

$$P_{ControlSignal} = error * P_{gain}$$

$$error\ sum = \sum error$$

$$I_{ControlSignal} = error\ sum * I_{Gain}$$

$$\Delta = Control\_Variable - Control\_Variable_{Last\_Bufferd}$$

$$D_{ControlSignal} = D_{Gain} * \Delta$$

$$Duty\,Cycle = Bias + \left(P_{ControlSignal} + I_{ControlSignal} - D_{ControlSignal}\right)$$

The duty cycle is bounded by the range 0 - 100 because duty cycles cannot be generated outside of these bounds. There is additional logic within the routine to avoid integrator wind-up and allow variable D periods to calculate Δ.

7.8. Maintenance Routine

Maintenance mode (MM) is implemented using the system's LCD display. The MM allows control of the system outputs, variables, and counts. When in MM, a global flag accessible to the whole system is set. This flag allows the system to perform certain actions and inhibits all faults (except ROM Checksum).

7.9. Graphic User Interface

The user interface provides viewing and adjustment of the laser operating parameters. The firmware is responsible for outputting all text and images to the LCD controller as well as monitoring the touch screen buttons. The GUI design is implemented in three layers:

1) Hardware Drivers = (Module Names = DispDrvr, Tch_scrn) LCD Display and Touch Screen dependent functions.
2) GUI Driver = (Module Name = Buttons) Consists of the H/W drivers as well as this module. The Buttons module functions create and destroy buttons, and respond to button events: OnDraw OnPress, and OnRelease.
3) GUI - Laser Specific Code = (Module Names =ScrnOper, ScrnMsgs, ScrnLVM, bitmaps) These modules contain the code that is specific to each laser. The design of the Primitives and GUI module should be such that they can be used in all previous and new laser designs using the two supported display panels without any changes.

There are some known design limitations:

- The 68hc12 uses banked memory to allow access above 64k. This poses a problem with bitmap data. Because of the number of languages supported, the bitmap messages stored in ROM are substantial (currently 1 bank [16k] is reserved per language). A design decision was made to place the DispDrvr code in the last bank. This allows the display code to be accessed from any other bank (bank architecture allows near accesses to the current bank, and the last bank).
- A problem can occur using the TextOut() function when displaying text strings to the screen. If TextOut() is called from bank 0, with the text string located in Bank1, it won't get displayed properly. It will display "?" (indicating unsupported character), and sometimes, the system would crash. This occurred with the P/N & Revision strings. The Compiler/Linker doesn't pick this up.
  Example:
  A call to "TextOut(layer, x, y, &font_8pt, fw_part_num)" is located in bank 1, and the string "fw_part_num" is located in Bank 0. The TextOut() call (bank 1) would execute the actual function code (bank 7), that accesses a far pointer to the needed font (bank 2). The text string (fw_part_num) is pointed to by a near pointer (simplifying using normal char strings). The TextOut() call can only access near variables in either it's bank 1, or from the last bank. So, the pointer to the text will access the incorrect data.
  Workaround - The text string used must be located in either the bank the TextOut() call is made from, or from bank 7.
- There are issues with "const Far ptr arrays". This can be fixed by casting BMP*'s to ulongs, and then back before using.
- The DispDrvr assembly code has DPAGE references. These are an artifact of using "far const's", even though it doesn't really support them. It is undetermined if this will be a problem with the next compiler rev.

7.9.1. Touch Screen (H/W driver)

The TS Calibration routine allows the calibration of software parameters to increase the accuracy of the mapping of the TS overlay with the display. The routine is entered automatically when the laser is powered on if no TS calibration parameters exist in BBRAM or if the parameters stored in BBRAM are out of range. The routine is also accessible from a button in LVM.

There are two functions (TS_GetX,TS_GetX) that are used to get the "raw" and scaled x and y position being touched from the touch panel. The "raw" position is an inverted 8bit A/D value to change the TS origin to the Upper Left. The "scaled" values are the raw values scaled to the display size in pixels (x max = 320, y max = 240) and also includes the Touch Screen calibration factors.

The TS is calibrated and verified by a series of touches to the TS within specified regions designated by boxes drawn on the display. The size and location of the calibration boxes and the tolerances of the verification boxes are all pre-defined.

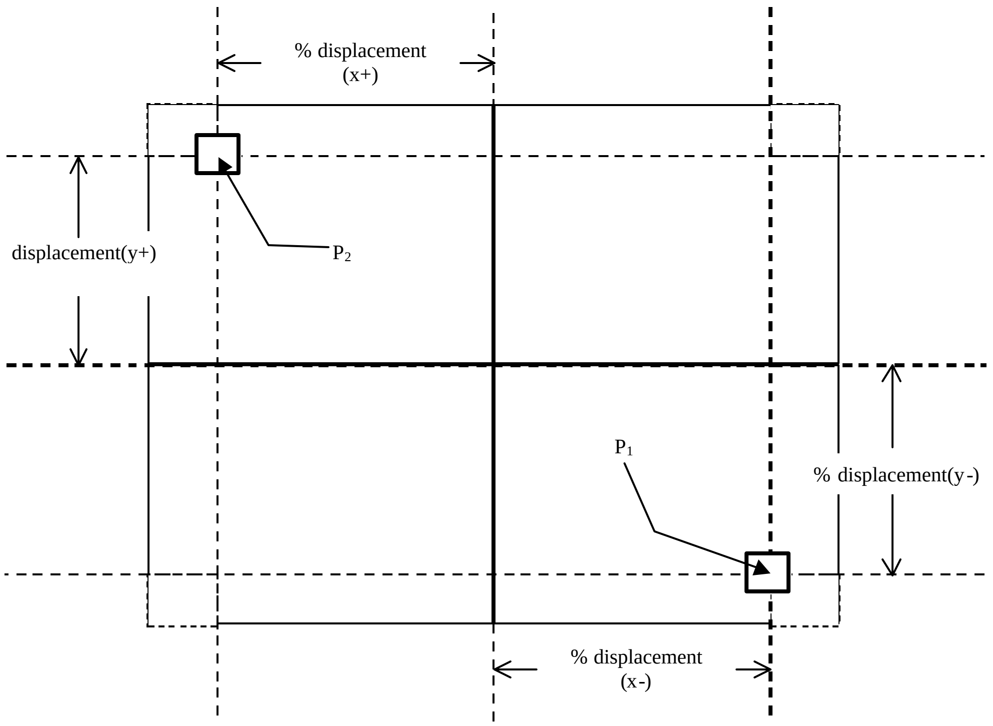


Figure 2: TS Calibration

The calibration requires 2 touches in order to determine the calibration values to proceed with the pixel_value calculation. The user touches the center of a box with a small blunt-tipped object. This box will then disappear and the user will touch the center of another box appearing in the opposite corner of the screen.

The position where the boxes will be displayed will be determined by a *%displacement* value for both the x and y axes of the LCD. The *%displacement* is a percentage distance from the center of the screen in both the x and y directions.

With the data acquired from the 2 points of the TS calibration, the calibration values are calculated for determining all *pixel_values*$_{xy}$

It is possible to calibrate the touch screen incorrectly, by not touching within the calibration boxes or a faulty touch screen. To ensure the correct values have been calculated, a calibration verification section has been created. Two verification boxes will be drawn following the completion of the ts_calibration phase. The verification boxes will be drawn at predetermined positions on the screen. The following conditions must be met in order for a calibration to be verified:

- A touch of a verification box will is sampled a predetermined number of times.
- For each sample obtained, the pixel_value will be calculated in both x and y directions. The calculated value must be within the bounds of the center of the validation box ± a window value. The window value = percent of Max_x or Max_y
- To successfully verify the TS calibration, for all samples taken in both x and y axes, a percentage of these values must be within the bounds of the window.

In order for a verification to be successful, both verification boxes must be properly verified. If the verification process doesn't prove the touch screen calibration was successful, default values are loaded and used to allow laser use.

7.9.2. LCD Display Driver (H/W driver)

The DispDrvr module has functions that interface with the LCD controller (SED1335) to display text and graphics. The controller supports a display image consisting of multiple 'layers' that can be updated independently and combined interactively. The first layer display all graphics and text, layer 2 is used for button pressed or highlighted, and LCD flashing, and layer 3 is not used. The ability to quickly update a small area of a single layer enables the GUI to create quick tactile feedback to the user.

There are 3 different implementations of LCD Controllers on Candela products. This display driver is intended to operate with any of them. However, the current design doesn't support interrupts (IRQ). This is not expected to be a difficult task. The MGL is set up to read/write the LCD display data polling the 1335 busy signal.

- VBeam -> LCD display - no data/status read – no IRQ
- SBeam -> EL display – data/status read – IRQ – Display has faster persistence, so shows changes as flicker

- CBeam/MGL -> LCD display – data/status read – no IRQ

The DispDrvr supports the following display primitives:

- LCDInit() – Initializes the LCD Controller
- FlashLayer() – Uses the LCD controller to flash the selected layer (1 of 3), at a specified rate
- DrawVLine() – Draw a vertical line in specified color
- DrawHLine() – Draw a horizontal line in specified color
- DrawBox() – Draw a box outline in specified color
- DrawFilledBox() – Draw a filled box in specified color
- DrawBmp() – Draw a bitmap at a specified location
- TextOut() – Display a text string using either 8pt or 10pt fonts (Arial bold)
- GetStrDim() – Determine the width in pixels of a text string

There are some efficiency enhancements that might be considered in the future:

- Objects should be byte aligned. Otherwise, the OutputLCDLine() function will "OR" the first and last byte with video memory to preserve those pixels.
- Manage LCD cursor position locally. Then only need to look at local copy of cursor position, instead of going out to display driver to get actual position.
- Draw V line can do multi pixel width. This would be more efficient than reading the surrounding video bits, multiple times so they don't get overwritten– reading potentially same byte.
- OutputLCDLine is used by everything – minimize loop data output loop timing.
- ShiftLine() is very time consuming. This is used by any object that's not byte aligned. Optimization of this is as important (or more) as optimizing OutputLCDLine().
- ShiftLine() could pass in int, and func could clear upper byte.
- Button bitmaps could include a released frame so when a button is drawn, it doesn't have to draw the button released frame as well.

7.9.3. Button Management

The Buttons module contains the functions necessary for button creation, bitmap an text drawing, and button event processing, button timers, and button group support.

There are a number of functions used to manage each button's properties. Each created button has an associated ID that is used by other button functions.

- CreateButton() – creates a button, sets the default bitmap, sets the button events, and returns an ID
- DrawButton() - Draws the specified button on the screen as not pressed (disabled)
- SetButtonBMP() - Sets the specified button's bitmap
- SetButtonText() - Sets the specified button's text
- SetButtonEnable() – Enables or Disable the specified button for presses
- IsButtonEnabled() – Tests if specified button is enabled
- SetButtonEvents() – Sets the specified button's events (OnPress, OnRelease, OnDraw)
- MoveButtonPosition() – Moves the button to a different location

The ScanForButtonPressed() function continuously polls the TS for changes. When the TS is being touched, the system calculates a pixel location and determines if the touched position maps into the boundaries of any of the currently enabled buttons. When three consecutive readings map into the same button, the OnPressEvent() function associated with the button is executed. The button is drawn as pressed, and a key click is heard. This function also calls the OnReleaseEvent() function associated with the pressed button, and deselects the button image when released.

There are two functions used to draw the button as pressed or released:

- ButtonShowDepressed()
- ButtonShowReleased()

There is one function that provides limited support of group buttons (Standby/Ready for example). A group is defined, and this function will handle the inversing necessary when one button is selected, and the others should deselect.

- SelectGroupButton() - Selects (inverts color) a specified already drawn Group Button

There are three timer functions used in the buttons module:

- UpdateButtonTimers() - Services all button releated timers (called on a periodic basis)

- SetButtonRepeatTime() – Sets a predefined repeat ramp (No, Slow, Med, Fast, Long) for the specified button. Some repeat rates increase as the button is held. This sets the rate that the OnPress event is called when the button is held down. The long press is also supported using this function, and causes the button to be drawn released after a determined long time.
- GetButtonPressedTime() – returns the length of time the button has been pressed. This can be used to determine an extended button press (Next press for 2 secs to enter LVM), though the repeat time can be also used for this function.

7.9.4. Laser Operation Screens

This module contains the MGL laser's screen specific code. There are functions to display each of the screens:

- DisplayCandelaScrn() – Displays Candela Screen
- DisplayMainScrn() – Displays Main Screen
- DisplayWarmupScrn() – Displays Warmup Screen
- UpdateWarmupSts() – Updates Warmup screen with % complete

There are other functions that display a portion of each screen, to allow quicker screen update:

- DisplayTopCmdBar() – Displays System State section
- DisplayBtmCmdBar() – Displays Purge, Cal, Next buttons section
- DisplayBtmLasing() – Displays Flashing Lasing section
- DisplayMidMain() – Displays Main parameter section
- DisplayMidOptions() – Displays Options parameter section
- DisplayMidMsg() – Displays message section
- DisplayMidWarmup() – Displays Warmup section

This module provides the event functions for the following buttons:

Main Screen Button Events

Standby
  OnPress – Clears all faults and goes to Standby state
Ready
  OnPress – Puts system into Ready state
Cal
  OnRelease – Starts a CAL. If pressed for > 2 seconds, it does a Full CAL
Next
  OnPress – If pressed > 2 seconds, it goes to LVM
  OnRelease – Toggles between Main and Options screens
Purge
  OnPress – If HP not in CP, Sprays for PreSpray + PostSpray time
Up
  OnPress – Dependent on the button that is selected. Normally, it increments the selected button's parameter
  OnRelease – If Pre or Post spray is selected, it updates the canister pulse count
Down
  OnPress – Dependent on the button that is selected. Normally, it decrements the selected button's parameter
  OnRelease – If Pre or Post spray is selected, it updates the canister pulse count
MainParGrp
  OnPress – If Fluence, Delay, PreSpray, PostSpray, or RepRate is pressed, highlights button
Fluence
  OnDraw – Displays the fluence button's parameter
RepRate
  OnDraw – Displays the Rep Rate button's parameter, and also SPM or MPM bitmap
PCount
  OnDraw – Displays the Patient count parameters
  OnPress – Resets Patient Pulse Count and LVM's Std Dev Count
PreSpray
  OnDraw – Displays the PreSpray button's time parameter
Delay
  OnDraw – Displays the Delay button's time parameter
PostSpray_DrEv()
  OnDraw – Displays the PostSpray button's time parameter
Canister
  OnDraw – Displays the Canister button's count parameter
  OnPress – Resets Canister Pulse Count

Options Screen Button Events

OptnParGrp
  OnPress – If Language, Trigger, or Aim Beam is pressed, highlights button
Language
  OnDraw – Displays the language button's language bitmap
Trigger_DrEv()
  OnDraw – Displays the trigger button's bitmaps
Aim_Beam_DrEv()
  OnDraw – Displays the aiming beam button's parameters

LVM Screen Button Events

TouchScr_PrEv()
  OnPress – Initiates a touch screen calibrate
Reset_Sigma_PrEv()
  OnPress – resets the LVM std dev statistics

7.9.5. Messages

The message screens enable the system to display instructional, system status, or warning messages. Messages appear in the central area of the screen and are displayed primarily as bitmaps, with some exceptions for additional text.

The message screen displays up to three buttons allowing the user to respond to the message on the screen. These buttons are usually used to cancel a procedure or supply acknowledgement of the message being displayed.

7.9.6. Bitmaps

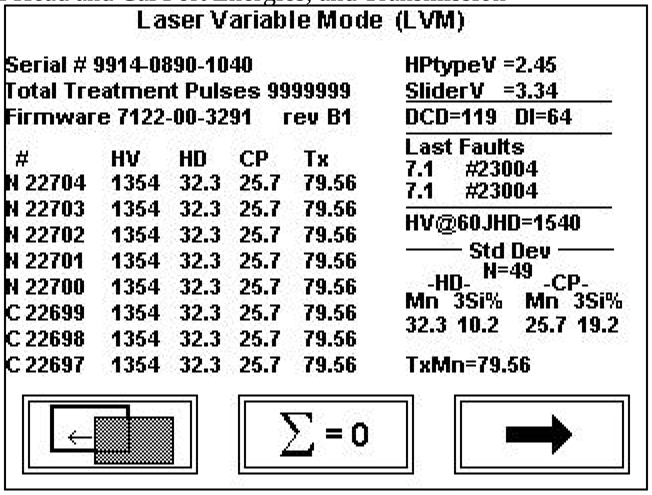
The text fonts supported are Arial Bold 8pt, and 10pt. Text and bitmaps are created in MS Paint using a 320x240pixels screen size. The following method is used for converting fonts (or any bitmaps into hex arrays):

1) In MS Paint, create a tall text box with characters down in a row.
2) Move text box so that it is at the left edge of bitmap.
3) Change attributes so that width is 16 pixels.
4) Flip picture vertically
5) Invert color
6) Run bmp_conv “directory” “outputFilename” (doesn’t seem to work on Win95) – this converts complete directory. Example: “bmp_conv . bmp.c” to create bmp.c file in current directory

7.9.7. Laser Variable Mode

The laser variable menu allows the user to monitor critical system parameters. Laser Variable Mode is entered by an extended press of the NEXT button. There are buttons for Touch Screen calibration, and Std Dev reset.
The key variables are:

- Firmware P/N and revision
- Laser Serial Number
- Total Treatment Pulses
- Laser pulse data for the last 8 pulses
- HP and Slider voltages
- DI pressure and DCD temperature
- Last 2 faults received
- Estimated HV at 60J Head energy
- Standard Deviation of Head and Cal Port Energies, and Transmission



Laser Variable Mode (LVM)

Serial # 9914-0890-1040
Total Treatment Pulses 9999999
Firmware 7122-00-3291 rev B1

#	HV	HD	CP	Tx
N 22704	1354	32.3	25.7	79.56
N 22703	1354	32.3	25.7	79.56
N 22702	1354	32.3	25.7	79.56
N 22701	1354	32.3	25.7	79.56
N 22700	1354	32.3	25.7	79.56
C 22699	1354	32.3	25.7	79.56
C 22698	1354	32.3	25.7	79.56
C 22697	1354	32.3	25.7	79.56

HPtypeV =2.45
SliderV =3.34
DCD=119 DI=64
Last Faults
7.1 #23004
7.1 #23004
HV@60JHD=1540
Std Dev
N=49

-HD-		-CP-	
Mn	3Si%	Mn	3Si%
32.3	10.2	25.7	19.2

TxMn=79.56

Table 3: System variables displayed in LVM

APPENDIX A. PULSE WIDTH MODULATION WITH THE HC12

Pulse width modulation (PWM)allows control of the output power of a system with 2 discrete states: on and off. This is accomplished by creating a **duty cycle**: manipulating the percentages of system on-off time with respect to a standard period, T_S. Pulse Width Modulated functionality can easily be accomplished using the output compare and timer capabilities of the 68HC12.

This is accomplished using the timer registers of the HC12 in conjunction with I/O lines of PORTT. For this application it is necessary to enable the output compare capabilities of the HC12 with selected I/O lines of PortT. The architecture of the HC12 allows synchronization of the state changes of all selected lines on PortT with a successful output compare of channel 7 timer. These actions are independent of the output compare actions selected to occur with the pins and their respective timers.

Timer channel 7 is set up so that a successful output compare will occur when the timer overflows, i.e. 16-bit count equals 0. This will be considered the beginning of a standard period, T_S. At this point, all selected output pins will be set to a high state, see timer 7 event of *Figure 3*. Each I/O line being used for PWM will have an associated timer. When a successful output compare occurs for the respective timer, one of the following actions will occur:

- Toggle of output line.
- Set line to logic level Low.
- Set line to logic level High.
- No action.

To accomplish what we are trying to do requires the inputs being pulled low state when the second timer expires, Timer events 5 & 6 of Figure 3. The set duty cycle will remain unchanged until the timers are changed or output compare is deactivated. The % duty cycle equals the on time w.r.t. the period time T_S.

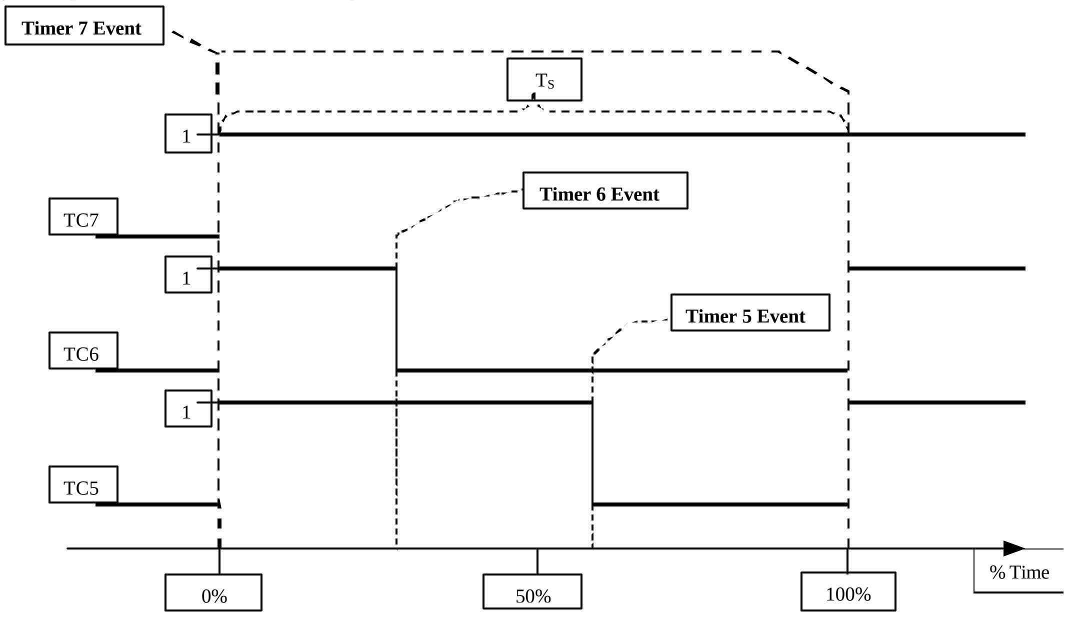


Figure 3- Output Compare Events

The following internal registers are used to initiate PWM.

- TIOS – allows corresponding pin on PortT to be set as Output compare.
- OC7M – allows corresponding pins to be changed on a channel 7 output compare event.
- OC7D – if set by above register, output lines assume the logic levels assigned by bits in this register on a channel 7 output compare event.
- TCTL1 & TCTL2 – set action to occur when output compare occurs on timer associated with pin.
- TCNT – 16-bit System timer.
- TCx – When TCNT = TCx an output compare will be “successful” and channel will perform action specified by TCTLx.

The values each TC_X should be set to are determined by the percentage results generated by a separate calculation.

APPENDIX B. DI PWM CONTROL ANALYSIS

CANDELA Memorandum

To: Cbeam Design File

From: Adam Best

Date: Tuesday, July 29, 2003

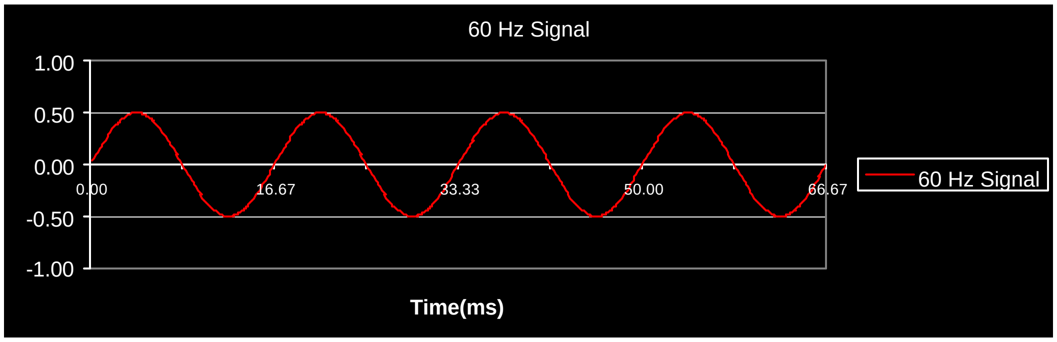
Re: DI Control Data

To PWM or not to PWM that is the question.

Well it looks like this question has been answered. Temperature regulation of fluid temperature on the Vbeam was simple and effective. To regulate fluid temperature at 60.0°C the heater would be on when the temperature T < 59.5°C. The fan would turn on when the temp erature was above 60.5°C. This proved effective at regulating temperature.

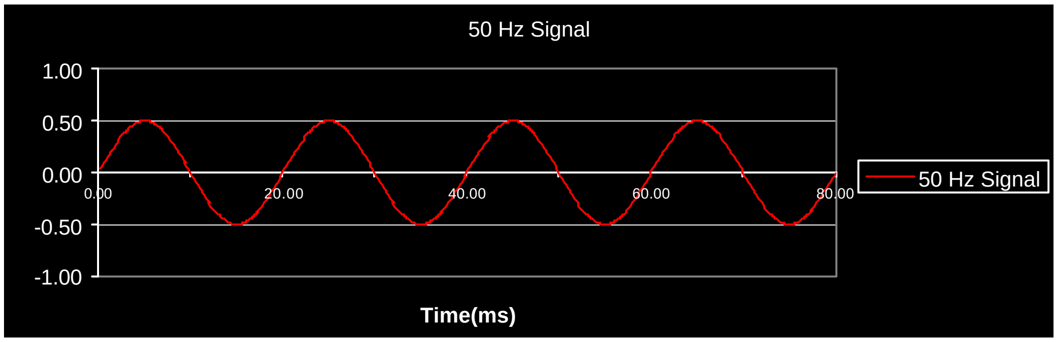
For the Cbeam, I had initially imagined the DI heater controlled be a variable duty cycle that would be calculated by a PI routine. In my imagination everything runs off of DC, in reality it does not. The triac, which controls switching of the DI heater on-off state, only allows the DI heater to change state at the zero-crossing of the AC signal. Result, you are left with a discrete number of duty levels with the number of levels dependant upon the AC frequency.

@ 60 Hz Period: $T_{60Hz} = 16.67ms$ Zero-Crossing Period: $T_{ZC\ 60} = \left(T_{60Hx} / 2 \right)$ = 8.335 ms



Graph 1 - 60Hz Signal

@50 Hz Period: $T_{50Hz} = 20.0$ ms Zero-Crossing Period: $T_{ZC\ 50} = \left(T_{50\,Hz} / 2 \right)$ = 10.00 ms



Graph 2 - 50Hz Signal

Full Cycle length of PWM: $T_{CYCLE} = 2^{16} * 1(\mu s) = 65.536$ ms .

#Duty Levels at 60 Hz : $\#DL_{60} = T_{CYCLE} / T_{ZC\ 60} =$ floor (65.536 / 8.335) = 7 non-zero states

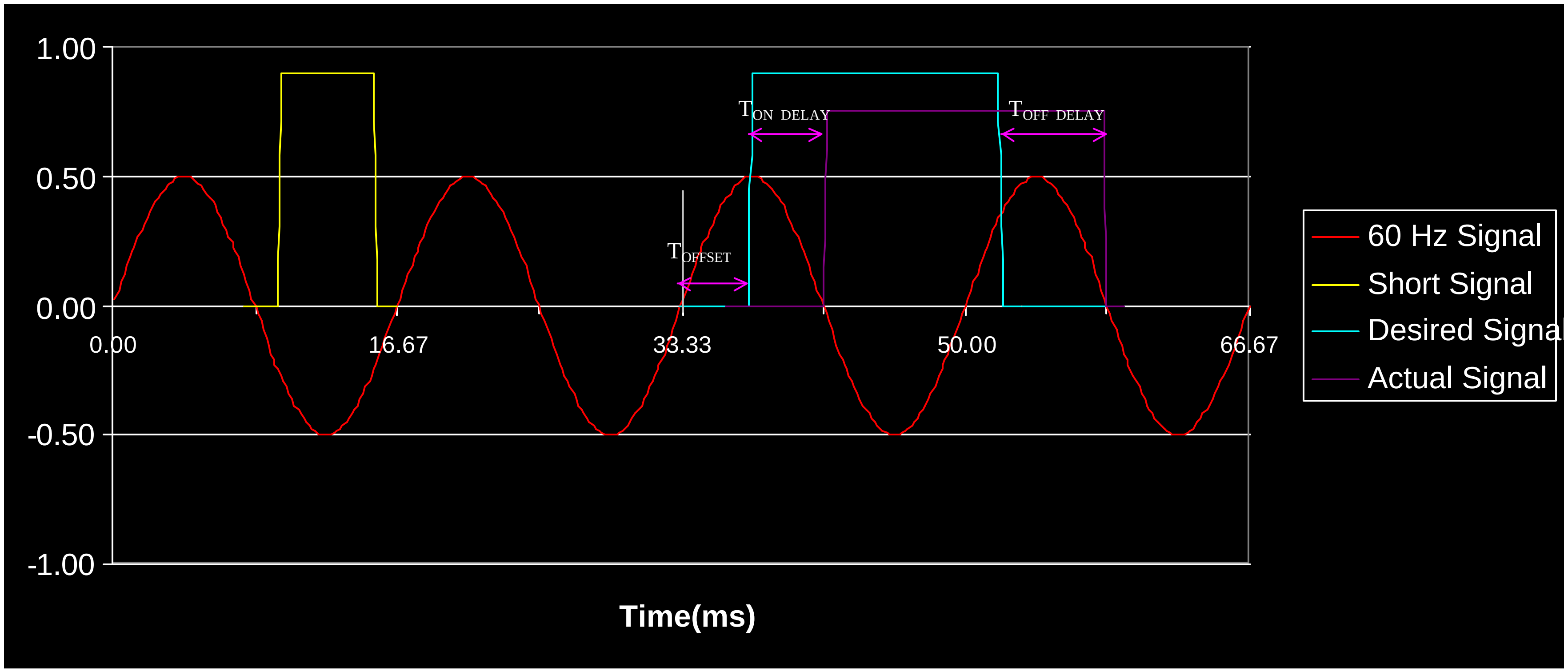
#Duty Levels at 50 Hz : $\#DL_{50} = T_{CYCLE} / T_{ZC\ 50} =$ floor (65.536 / 10.00) = 6 non-zero states

For both frequencies there are 2 additional states:

- OFF state, where the DI heater is NOT turned on during period T_{CYCLE} . This state occurs when
  - Duty Cycle = 0%

    DI heater pulse width < T_{ZC} and the entire length of the pulse occurs between 2 zero-crossings. (See
  - **Graph 3** point A)
- ON state, where the DI heater is on during period T_{CYCLE}. This state occurs when:
  - Duty Cycle = 100%
  - DI heater pulse width > (#DL*T_{ZC}) and entire pulse length covers (#DL+1) zero crossings.

Additional complications exist with implementing PWM to control the DI heater. The most important being able to accurately determine how long the DI heater will be on. This uncertainty exists because the zero-crossing period divides into the period $T_{60\ Hz}$ leaving a remainder. **Graph 3** - displays how this can occur.



Graph 3 - DI Control signal with 60 Hz AC

T_{Offset}: Time offset from zero-crossing to the beginning of the 65.536 (ms) PWM period of µC

T_{ON_Delay}: Time from DI "turned on" by µC until next zero-crossing.

T_{OFF_Delay}: Time from DI heater output "turned off" until next zero-crossing.

Desired Signal $PW_{DESIRED}$= duty cycle * T_{CYCLE}

Actual Signal $PW_{ACTUAL} = T_{ZC} \cdot Floor\left[\frac{T_{offset} + PW_{DESIRED}}{T_{ZC}}\right]$

How does this remainder affect the output signal?

In this example the **Desired Signal** pulse width = 14.5 ms.. However, the **Actual Signal** to the DI heater = $2 \cdot T_{ZC.}$ Ideally, a signal with pulse length $< 2 \cdot T_{ZC}$ would be rounded down to T_{ZC}. However, this assumes the beginning of the T_{CYCLE} is synchronized with the zero-crossing. This is not the case.

Rather, there exists a time, T_{OFFSET}, between the time of the zero-crossing and the beginning of a PWM cycle. T_{OFFSET} is a function of the T_{ZC} and the amount of time the system has been on. Therefore, the **Actual Signal** pulse length is dependant not only on the desired duty cycle, but a combination of the desired duty cycle and the time T_{OFFSET}. The result of the uncertainty of the **Actual Signal** pulse length is a varying error:

$$\%Error = \left(\frac{PW_{ACTUAL}}{PW_{DESIRED}} - 1 \right) \cdot 100$$

Although it is possible, it makes little sense to devote any significant time trying to tailor the controller to regulate the DI heater using PWM if the controller itself is not reliable. The current method of fluid temperature regulation works well enough for the application.

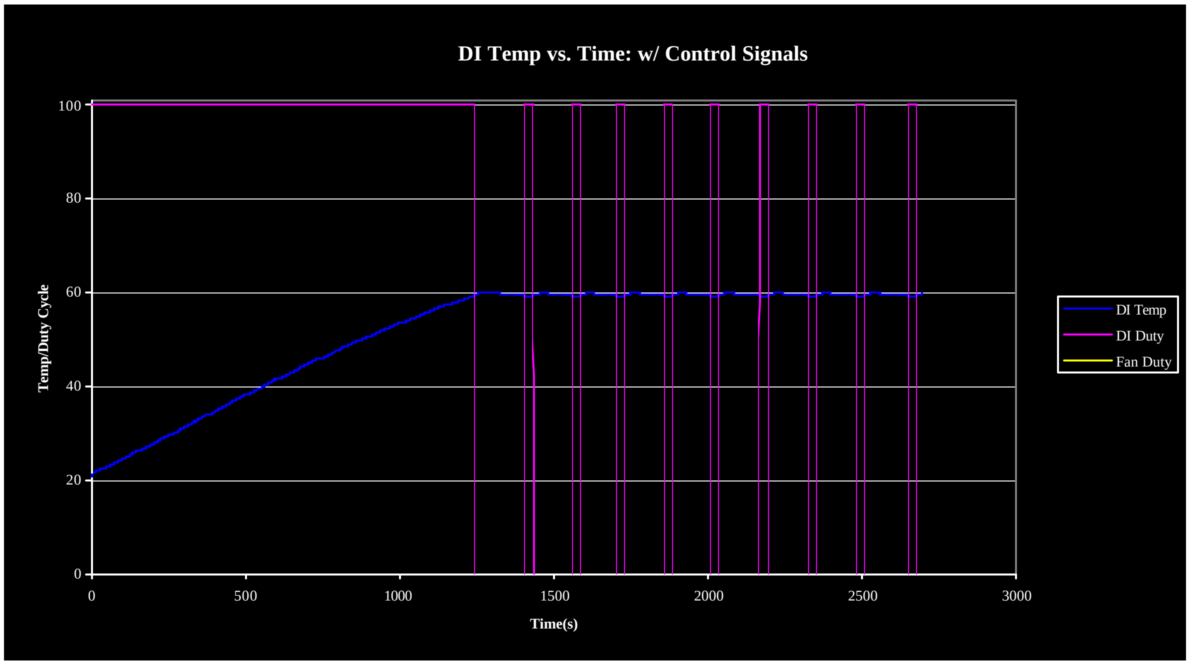
The "Current Method" Data

The current method of DI heater control is quite simple:

- Temperature < 59.5°C; the fluid heater comes ON.
- Temperature > 59.5°C; the heater is OFF.

The following data was obtained using these guidelines for controlling the heater temperature:

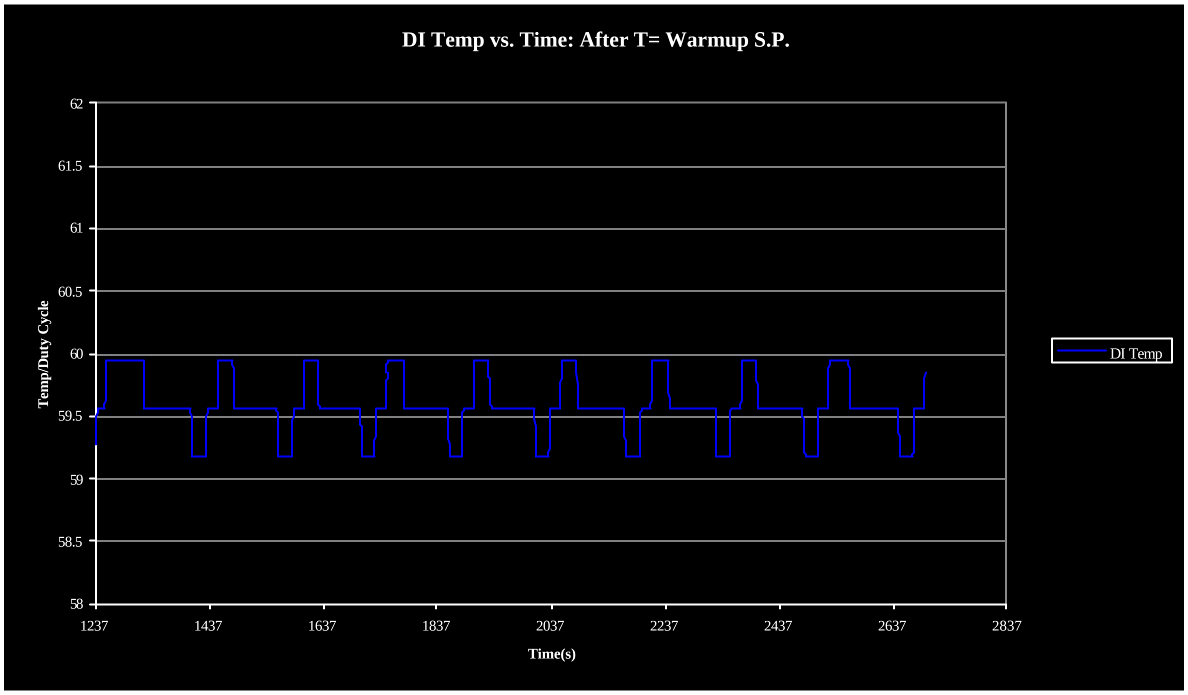
Initial Temperature:	**20.75°C**
Warm-up Setpoint:	**59.5°C**
Time to reach Warm-up Setpoint:	**1240 s**; 20 min, 40 sec



Graph 4 - DI Temperature vs Time

After Reaching Warmup Temperature:

Average Temperature after warm-up: **59.58°C**



Graph 5- DI Temp vs Time after system has warmed up. Standard DI control.

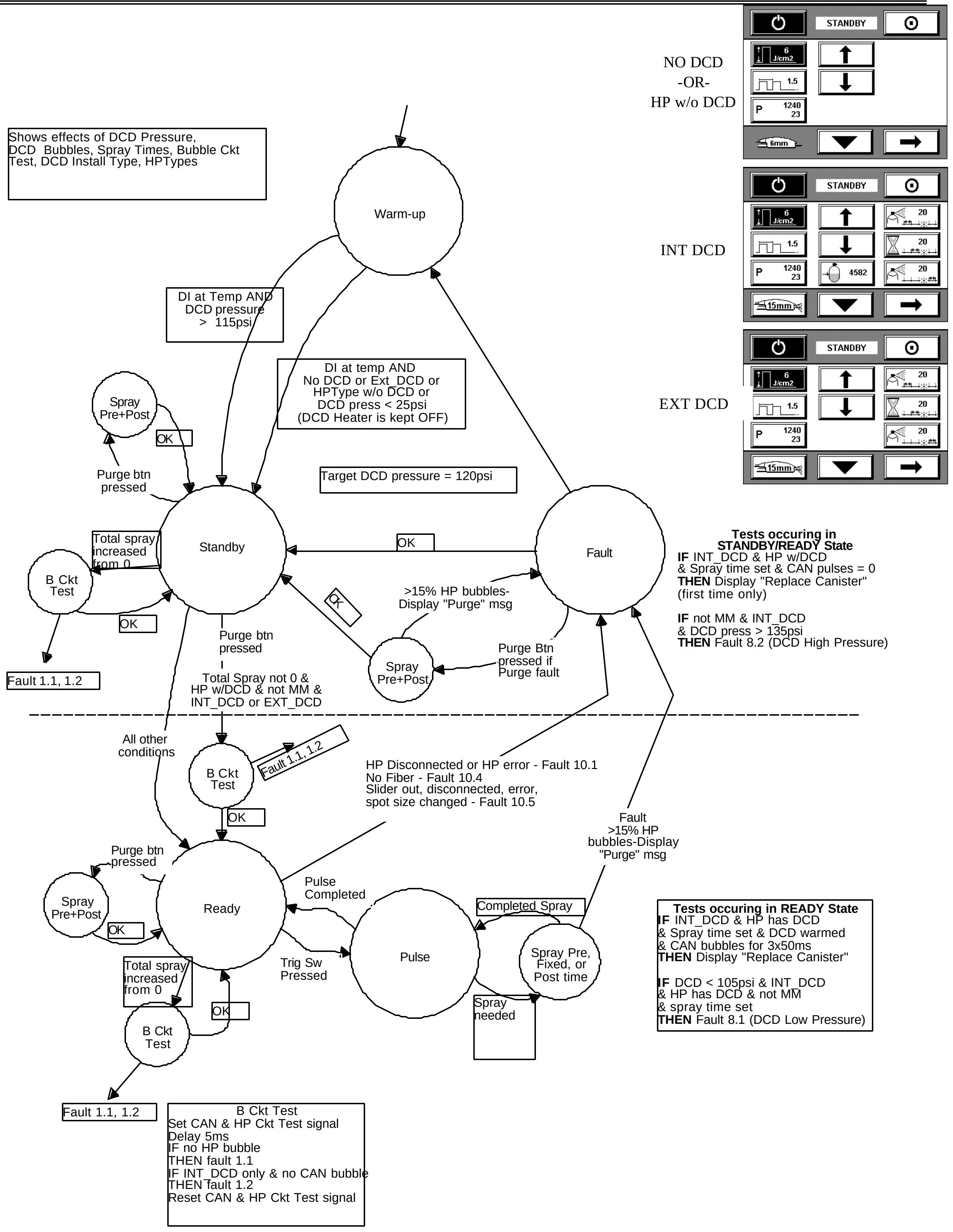
Graph 5 shows how the control manages to keep the DI fluid temperature near the set point. Although the average temperature after warm-up, tends to fall below the set point, the error is within the ± 1° C specification for the system. Therefore, this method of control will be implemented on this system.

Cycle	Heater Timer ON (s)	Heater Timer OFF (s)	Period Length (s)
1	27	164	191
2	27	122	149
3	26	117	143
4	26	125	151
5	26	125	151
6	28	129	157
7	26	130	156
8	27	125	152
9	27	139	166
Average	26.67	130.67	157.33

APPENDIX C: MGL HP VOLTAGE RANGES

The MGL operates with five different delivery systems. The CPU recognizes a delivery system based on an analog voltage set by a potentiometer in the HP. The following Table lists the delivery system's range recognized by the CPU. Note that the decoding is +-5bits or 10% (whichever is greater.)

Type	Ohms	Volts	Bit
Disconnected	~	0.000	0
	20545	0.231	12
Error	19733	0.240	13
	12595	0.365	19
	9104	0.490	25
Error	9083	0.491	26
	7030	0.615	32
	5657	0.740	38
12mm LE	5648	0.741	39
	4680	0.865	45
	3951	0.990	51
12mm LE w/DCD	3945	0.991	52
	3384	1.115	58
	2932	1.240	64
12/15/18mm	2929	1.241	65
	2513	1.384	72
	2174	1.527	79
Error	2172	1.528	80
	1857	1.691	88
	1597	1.854	96
12/15/18mm w/DCD	1595	1.855	97
	1332	2.056	107
	1115	2.257	117
Error	1114	2.258	118
	886	2.517	131
	701	2.776	144
6/8/10mm	701	2.777	145
	516	3.094	161
	383	3.372	177
6/8/10mm w/DCD	382	3.373	178
	221	3.785	197
	102	4.160	216
Error	99	4.170	217
	0	4.900	255


NO DCD
-OR-
HP w/o DCD
INT DCD
EXT DCD
STANDBY
6 J/cm2
1.5
P 1240 23
6mm
15mm
4582
20
Shows effects of DCD Pressure, DCD Bubbles, Spray Times, Bubble Ckt Test, DCD Install Type, HPTypes
Warm-up
DI at Temp AND DCD pressure > 115psi
DI at temp AND No DCD or Ext_DCD or HPType w/o DCD or DCD press < 25psi (DCD Heater is kept OFF)
Target DCD pressure = 120psi
Spray Pre+Post
OK
Purge btn pressed
Standby
Total spray increased from 0
B Ckt Test
OK
Fault 1.1, 1.2
OK
Fault
>15% HP bubbles- Display "Purge" msg
OK
Spray Pre+Post
Purge Btn pressed if Purge fault
Purge btn pressed
Total Spray not 0 & HP w/DCD & not MM & INT_DCD or EXT_DCD
Tests occuring in STANDBY/READY State
IF INT_DCD & HP w/DCD & Spray time set & CAN pulses = 0 THEN Display "Replace Canister" (first time only)
IF not MM & INT_DCD & DCD press > 135psi THEN Fault 8.2 (DCD High Pressure)
All other conditions
B Ckt Test
Fault 1.1, 1.2
OK
HP Disconnected or HP error - Fault 10.1
No Fiber - Fault 10.4
Slider out, disconnected, error, spot size changed - Fault 10.5
Fault >15% HP bubbles-Display "Purge" msg
Purge btn pressed
Spray Pre+Post
OK
Ready
Pulse Completed
Completed Spray
Pulse
Spray Pre, Fixed, or Post time
Trig Sw Pressed
Spray needed
Total spray increased from 0
OK
B Ckt Test
Fault 1.1, 1.2
Tests occuring in READY State
IF INT_DCD & HP has DCD & Spray time set & DCD warmed & CAN bubbles for 3x50ms THEN Display "Replace Canister"
IF DCD < 105psi & INT_DCD & HP has DCD & not MM & spray time set THEN Fault 8.1 (DCD Low Pressure)
B Ckt Test
Set CAN & HP Ckt Test signal
Delay 5ms
IF no HP bubble
THEN fault 1.1
IF INT_DCD only & no CAN bubble
THEN fault 1.2
Reset CAN & HP Ckt Test signal

Mini-GL DCD Controls

APPENDIX E: MM DESCRIPTION

1.0 MAINTENANCE MODE OVERVIEW

Maintenance Mode is an interface to grant unabated access of system metrics and controls to Candela service personnel, technicians, and engineers. This extended set of functions and routines within system firmware enables service personnel to monitor data, set system critical values, and manipulate controls for system testing and diagnostics. With the Mini-GL we seek to incorporate the full Maintenance Mode interface into the system and allow complete Maintenance Mode access using the TSDP for both input and display of system critical information. Full Maintenance Mode interface capabilities enable service technicians to perform the following tasks:

- Test system outputs.
- Monitor Boolean inputs.
- Monitor input metrics.
- Set output states and output values.
- Access system statistics.
- Output system data to an external computer.

2.0 ADDITIONAL DEFINITIONS

The following are additional definitions used in this document:

MM:	Maintenance Mode firmware incorporated into sys
User:	A trained Candela service technician with MM access code.

3.0 DESIGN SPECIFICATIONS

As mentioned before, MM is primarily an interface. Many system processes are determined by and therefore has access to all system data that is to be output in MM. It is the goal of MM to provide access these data and controls, present them to the user, and allow manipulation of system controls by the user through an interface.
Many of the design specifications of maintenance mode are dependent on system limitations.

- All data must fit on a 320 x 240 display limiting icon detail, the size and legibility of fonts.
- 8-bit touch screen resolution limits the precision of touch recognition.
- An 8 Mhz processor controls the entire system, making it is important to minimize frequency and the scope of LCD updates.

Design of the MM interface should posses the following characteristics:

- Clarity
- Ease of use
- Robust Control of laser sub-systems
- Expandability
- Graphical Consistency – "The look and feel"
- Update of the MM interface should not compromise system performance
- Graphical User Interface is easily updateable.
- Ease of control – adhering to the *'Handheld User Interface – Ten Commandments*'
  "Thou shalt allow chubby fingers to operate a touchscreen"

4.0 MAINTENANCE MODE SCREENS

Traditionally, acquisition of laser data was performed using an external computer functioning as a dumb terminal or running proprietary software. Communication was accomplished using a serial cable using the RS-232 interface. The laser would establish connection with the external terminal and transfer formatted data to the terminal. Control of the system was accomplished by sending commands and formatted data from the terminal to the system.

Operating MM on an external computer enabled the use of a monitor with VGA resolution and in some cases colordisplay. However, the laser system is equipped with a half-VGA monochrome display. The decreased display size and reduced resolution make it impossible to present all necessary data on one screen. Therefore, it is necessary to group together relevant data and make the data available based on the frequency of use. This section details the MM screens implemented and the purpose each with respect to system control and data acquisition.

4.1. Maintenance Mode Access Screen

Previously an external computer was required to access MM. With MM in system firmware and fully operational without the need for an external computer or software, it is extremely important to restrict MM access to Candela Laser technicians, service personnel, and engineers. Improper use of the laser MM by the customer has the potential to injure the customer or harm the laser. Restriction of the MM access is accomplished with the use of the MM access screen.

The MM access screen displays a secure keypad allowing the user to enter an access command for entering MM. Candela service personnel should only know the access command.

If three unsuccessful access attempts are made or an access attempt is not completed within a set time period, the system will return to the normal system operation state. When the correct access code is entered, the system removes the keypad from the display, enters into MM, and sets a flag allowing the user to enter MM and from the normal operation mode without a password. This flag will remain valid until the system is powered down.

4.2. Multi Purpose Screen Regions

As mentioned earlier, all of the information and controls in maintenance mode cannot be implemented on one screen. However certain controls must always be present to the MM user. These controls should:

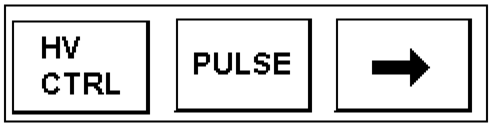
- Inform the user of the operational state of the laser state.
- Permit access to all MM screens.
- Permit return to “Normal” system operation.

4.2.1. Navigation from one screen to the next While in MM, certain controls must always be displayed to enable the user to navigate the MM screens. As an issue, it is important that the MM user always be aware of the laser state.

4.2.2. Screen Control Region

The requirements of the Screen Control region are simple: display and allow direct access to the user all available MM screens. To meet these specifications, the region was designed to consist of three buttons. The following table lists the region buttons and the functions of those buttons.

BUTTON	FUNCTION
MM Screen Scroll	Scroll through the list of MM screens. Redraw the names of the buttons directly accessible by MM Screen Selection buttons 1 & 2.
MM Screen Selection 1	Display the MM screen name accessible with a button press. Draw the MM screen displayed when the button is pressed.
MM Screen Selection 2	Display the MM screen name accessible with a button press. Draw the MM screen displayed when the button is pressed.

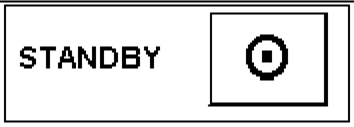


4.2.3. State Control Region

In MM, the user should always be informed of the system state and have control over it. Control of the laser system state in MM should be performed by the system in the same manner as normal operation mode. The State Control region should display the operational state of the laser (STANDBY, READY) and allow the user to control these states. Displaying the system state and providing a control to switch between the system states achieve this goal. While in MM the system states operate as they do in normal mode with the exception of not faulting when fault exceptions occur.

BUTTON	FUNCTION

MM State Button	Button calls the StateChange() routine when pressed. The icon on the button changes to reflect the state of the button when the system state change is complete.

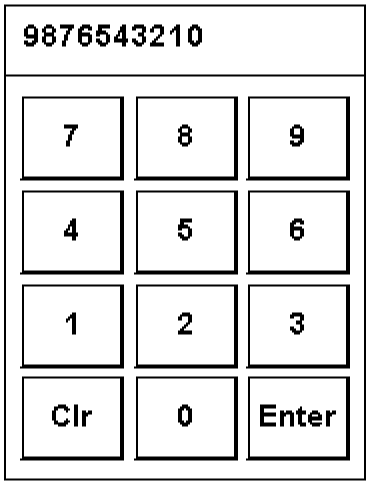


4.2.4. Keypad Region

The Keypad is used on multiple MM screens to allow user input of numeric values that are assigned to system variables. The keypad design consists of a TextLabel displaying the number entered, a button for each digit 0 - 9, a ‘Clear’ key to delete the last digit entered, and an “Enter” key allows the entered value to be assigned to the system that requested the number.

In order for the keypad to be used it must be initialized with the minimum and maximum values that can be entered. The keypad also can be set up to be “secure” so that for any digit entered, an asterisk is displayed. Once launched, the keypad disables input to all buttons of all visible screen regions other than itself, Screen control, State control.

BUTTON	FUNCTION
Numeric Inputs 0 – 9	Inserts the digit of the button pressed at the end of the value if new value is less than the maximum value.
Clear	Deletes the last number entered from the value displayed on the keypad.
Enter	Returns the value entered to object that requested it.



4.3. Toggle Screen

The Toggle Screen allows the user to monitor the state of inputs; toggle the state of outputs and control flags; monitor system metrics; and set the duty cycles of system controls. The information and controls of the Toggle screen are grouped together to allow the user to inspect all data pertinent to a subsystem.

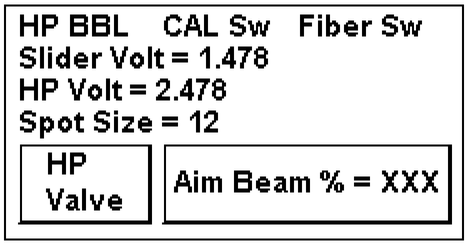
When in MM, the system is set up to update the toggle screen on a periodic basis to provide the user with current system information. For the most part values and inputs, data is only written to the screen when data has changed. This prevents excessive unnecessary display updates that could possibly interfere with system timing.

4.3.1. HP Info Region

The HP info region displays all data and controls relating to the delivery system. These values can be used to verify correct identification of the delivery system as well as the status of all electrical and optical connections with the system.

The following inputs and metrics associated with the delivery system are as follows:

INPUTS	HP Bubble	
	Calport switch state	
	Fiber switch state	
METRICS	Slider Voltage	(V)
	Handpiece Voltage	(V)
	Spot Size	(mm)



The following controls associated with the delivery system:

BUTTON	FUNCTION
HP Valve	A button press toggles the state of the HP Valve in the hand piece.
Aim Beam %	Launches the Keypad with Min Value = 0, Max Value = 100. The aiming beam duty cycle is set to the value returned from the keypad.

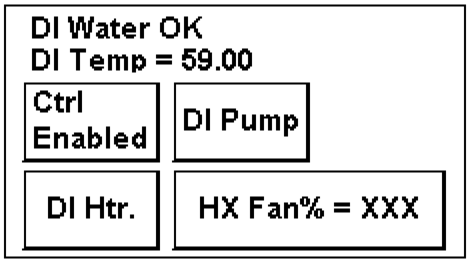
Along with controlling outputs, the control buttons also indicate the state or value of the associated control.

4.3.2. DI Info Region

The DI Info Region displays data and controls relevant to the DI subsystem. The region enables the user to verify operation of DI temperature regulation components (DI Htr, HX Fan), DI flow controls, and automated DI temperature regulation.

The following inputs and metrics associated with the DI system are as follows:

INPUTS	DI Water OK
METRICS	DI Temperature (°C)


DI Water OK
DI Temp = 59.00

The following controls associated with the DI system:

BUTTON	FUNCTION
DI Control Enable	On a button press, the temperature regulation is toggled (engaged or disengaged).
DI Pump	Button press toggles the output signal to the DI Pump
DI Heater	Button press toggles the state of the DI Heater when temp control is disabled.
HX Fan %	Launches the Keypad with Min Value = 0, Max Value = 100. The HX Fan duty cycle is set to the value returned from the keypad.

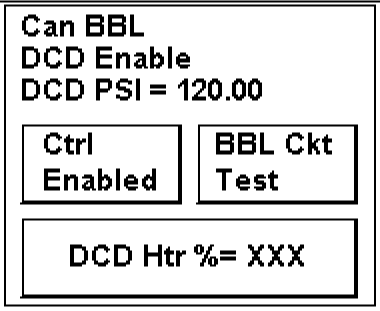
Along with controlling outputs, the control buttons also indicate the state or value of the associated control. Setting the DI Heater Control and the HX Fax duty cycle while DI Ctrl is enabled will cause a temporary change it the state and duty cycle respectively. However, the values will be reset by the control on the next iteration through the *DITempCtrl()* routine.

4.3.3. DCD Info Region

The DCD Info Region display all data and controls relevant to the control of DCD pressure and fault recognition. It is important to note that spray of DCD is considered a function of the delivery system.

The following inputs and metrics associated with the delivery system are as follows:

INPUTS	Canister Bubble Detection
	Internal System DCD Enabled
METRICS	DCD Pressure (p.s.i)



The following controls associated with the DCD system:

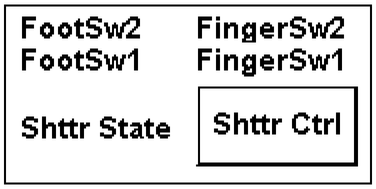
BUTTON	FUNCTION
DCD Control Enable	On a button press, the pressure regulation is toggled (engaged or disengaged).
Bubble Circuit Test	On a button press, the Bubble Circuit test is toggled (engaged or disengaged).
DCD Heater %	Launches the Keypad with Min Value = 0, Max Value = 100. The DCD heater duty cycle is set to the value returned from the keypad.

Along with controlling outputs, the control buttons also indicate the state or value of the associated control.

4.3.4. Laser Trigger Region

The Laser Trigger Region displays all input with respect to the triggering the laser and output energy. The laser inputs region displays the state of the inputs associated with the triggering of the laser.

INPUTS	Footswitch 1 & 2
	Fingerswitch 1 & 2
	Shutter State



The following controls associated with the Laser Trigger system:

BUTTON	FUNCTION
Shutter Control	On button press, the state of the shutter is toggled.

Along with controlling outputs, the control buttons also indicate the state or value of the associated control.

4.4. High Voltage Screen

The High Voltage Screen provides the user with information of the input voltage and output energy. The screen consists of three information regions and a Keypad for input of the HVPS charge voltage. The primary use of the screen is to enable the user to set the system target charge voltage, as well as monitor the actual charge voltage and the amount of energy.

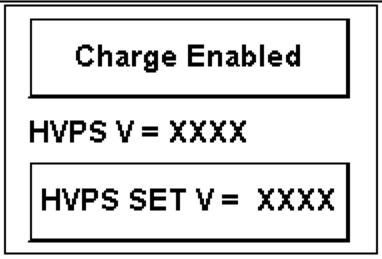
Like the Toggle screen, data acquired from the system and piped to the various regions of the High Voltage control screen only updated when the data has changed.

4.4.1. HVPS Region

The HVPS region provides the user with the pertinent information relating to the HVPS and enables the user to control the target charge voltage as well as the state flag that controls charging will occur prior to pulsing.

The following metrics associated with the HVPS region are as follows:

METRICS	HVPS Sample	(V)



The following controls associated with the DCD system:

BUTTON	FUNCTION
Charging Enable	On a button press, the charging enabled flag is toggled between enabled and disabled states.
HVPS Set	Launches the Keypad with Min Value = 0, Max Value = Max Charge Voltage. The target charge voltage is set to the value returned from the keypad.

Along with controlling outputs, the control buttons also indicate the state or value of the associated control.

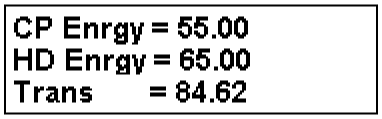
The Charging Enable button determines if the laser will charge before the next pulse. If the laser is already charged, the capacitor will retain its charge until the system is pulsed or the HVPS is disabled when entering the STANDBY state.

4.4.2. Laser Energy Region

The Laser energy region displays the output energy measured by system on the last pulse. These values aid service personnel in determining delivery system efficiency and isolate energy loss to specific delivery system components.

The following metrics associated with the HVPS region are as follows:

METRICS	Calport Energy	(J)
	Head Energy	(J)
	Transmission	(%)

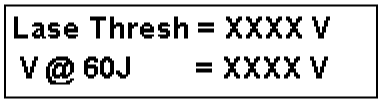


4.4.3. Laser Characteristics Region

The laser characteristics region outputs values acquired by the system on a full calibration. These values give service personnel an impression of the of laser head degradation.

The following metrics associated with the HVPS region are as follows:

METRICS	Lasing Threshold	(V)
	Voltage To Produce 60 J	(V)

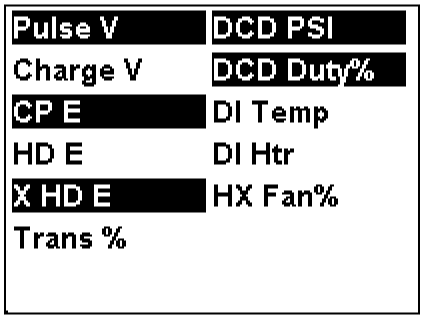


4.5. Pulse Data Selection Screen

The Pulse Data Selection and Pulse screens allow the user to setup MM to synchronize acquisition of system data with laser pulsing. The Pulse Data Selection screen sets up the data the Pulse screen will acquire when the laser pulses and the order in which the data will be presented. This screen is constructed of the following components.

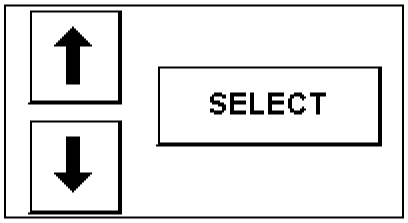
4.5.1. Pulse Selection Region

The Pulse Selection Region is a list of all data that can be selected for output on the Pulse Screen. When an index of the Pulse Selection region is selected for the pulse data screen, the index becomes highlighted. The order in which the data is selected from the Pulse Data Selection screen determines how the data will appear on the pulse screen.



4.5.2. Selection Control Region

The selection control region enables the user to scroll through the list of the Pulse Data Region. On an Up/Down arrow press, the label above or below the active label will become active and the last label will become de-activated. Pressing the Select button will add the data to a list of selected labels. All labels selected for display on the pulse data screen will be highlighted.



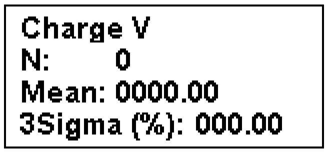
4.6. Pulse Screens

The Pulse screen is used to acquire and display a selected list of pulse data for each laser pulse. When the Pulse screen is entered, via the Screen Control menu, MM initializes the screen to acquire data from the fields previously selected on the Pulse Data Selection screen. Due to the limitations of screen size, only the first 6 selected data fields can be output. All of the remaining pulse data is available to the user if the user connects an output terminal to the system. The ordering of that data is dependent on the order of the data selected from the Pulse Data Screen.

4.6.1. Statistics Region

The statistics region acquires the data of the selected pulse data index. The region allows the MM user to determine the deviation of any of the selected Pulse data metrics.

METRICS	Selected Pulse Index
	N Pulse Count
	Mean
	3 Sigma Percentage (%)

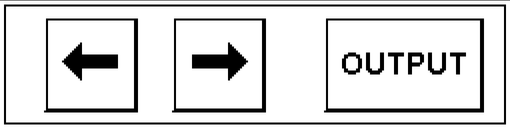

Charge V
N: 0
Mean: 0000.00
3Sigma (%): 000.00

All pulse statistics are reset when the left and right selection control of the selection control regions are pressed.

4.6.2. Selection Control Region

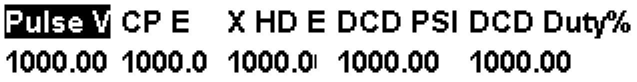
The selection control region of the Pulse Screen allows the user to select which pulse selection will be output to the statistics region.

BUTTON	FUNCTION
Left/ Right Selection Control	On a button press, the selected index is moved to the next available index of the pulse screen. In addition the statistics section is reset when the next button is pressed
Historic Pulse Update	On a button press, the most recent 7 pulses of the current Pulse Data setup, is output to the display.


OUTPUT

4.6.3. Pulse Data Region

The Pulse Data Region outputs data from the most recent laser pulse. The format of the data output



```
Pulse V CP E    X HD E DCD PSI DCD Duty%
1000.00 1000.0  1000.0  1000.00  1000.00
```

4.6.4. Historical Pulse Data Region

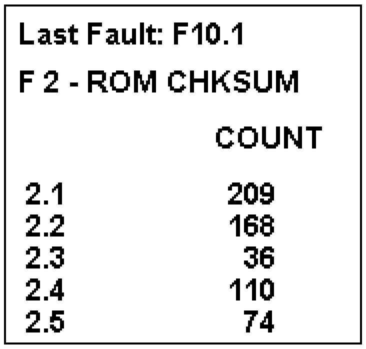
The Historical Pulse data section

4.7. Fault Screen

The fault screen displays fault count of all major fault and each occurrence of each sub-fault. The screen additionally displays the full fault number of the last fault to occur and allows the user to reset individual fault counts or reset the whole fault table. The following sections detail the components of the Fault Screen

4.7.1. Sub-Fault Log

The Sub-fault Log contains data about the occurrence of each sub-system sub-fault to aid Candela technicians, service personnel, and engineers in identifying and isolating system malfunctions. The Region also displays Last and current fault data.



```
Last Fault: F10.1
F 2 - ROM CHKSUM
              COUNT
2.1             209
2.2             168
2.3              36
2.4             110
2.5              74
```

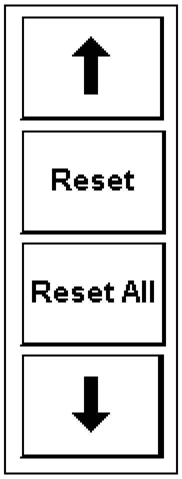
4.7.2. Total Fault Log

The total Fault Log displays the sum of all sub-faults of each major fault to give a general indication of the problems associated with each laser subsystems.

4.7.3. Selection Control

4.7.4. The Selection Control region controls scrolling through and resetting

BUTTONS	FUNCTION
Up/Down	Allows the user to view the sub-faults of major faults.
Reset	Resets the count of all sub-faults of the selected major fault.
Reset All	Resets the count of all sub-faults of for all major faults.



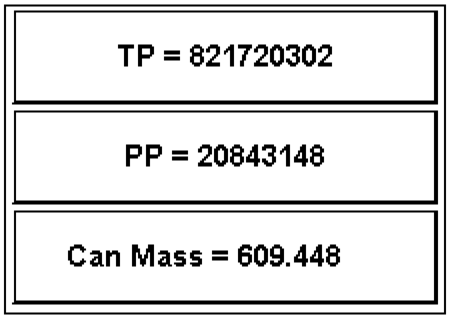
4.8. Counts Screen

The Counts Screen enables the user to set and reset historic values. The screen is composed of a keypad and the Count Update Region. All buttons on the Count Update region launch the keypad. The keypad disables access to the Count Update region while a value is being entered and returns the entered value to the button that launched the keypad.

4.8.1. Count Update Region

The Count region allows the user to update counts and values kept in BBRAM that are not cleared when cycling power or changing lasers software revisions. This region allows the user to input and update the following values:

BUTTON	**FUNCTION**
Total System Pulse Count	The total pulse count is a value that is reset in the factory and should not be updated otherwise.
Patient Pulse Count	The Patient Pulse count is the number of non-maintenance, non-calibration pulses logged on the laser.
DCD Canister Mass	The number of remaining cryogen pulses is determined using the spray settings and the assumed remaining mass of the canister


TP = 821720302
PP = 20843148
Can Mass = 609.448