| Data | PMC use | Existing board use | App use |
| --- | --- | --- | --- |
| Inlet temperature | * Monitor sensor for signs of failure * Log data to local storage * Send data to App | * Calculate “pad temperature” | * Display data (locked admin screen) * Send data to database |
| Outlet temperature | * Monitor sensor for signs of failure * Log data to local storage * Send data to App | * Calculate “pad temperature” | * Display data (locked admin screen) * Send data to database |
| “Pad temperature” | * Log data to local storage * Send data to App | * Control relevant reservoir temperature during treatment | * Display data (non-blinded GUI and locked admin screen) * Send data to database |
| Hot reservoir temperature | * Monitor sensor for signs of failure * Log data to local storage * Send data to App | * Send data to PMC * Control heating of hot reservoir during cold cycle | * Display data (locked admin screen) * Send data to database |
| Cold reservoir temperature | * Monitor sensor for signs of failure * Log data to local storage * Send data to App | * Send data to PMC * Control cooling of cold reservoir during hot cycle | * Display data (locked admin screen) * Send data to database |
| Heater temperature | * Monitor sensor for signs of failure * Monitor heater temperature and send command to stop heating if too high * Cut power to heater if software command fails | N/A | * Display warning if too hot (stop heating command sent) * Display data (locked admin screen) * Send data to database |
| SW revision (existing board) | * Read control board firmware version * Send data to App | * Send data to PMC | * Send data to database * Display data in info screen |
| SW revision (PMC) | * Send data to app |  | * Send data to database * Display data in info screen |
| Cold set point | * Send set point to control board * Send data to App | * Set target cold temperature | * Send data to database * Display data (non-blinded GUI and locked admin screen) |
| Hot set point | * Send set point to control board * Send data to App | * Set target hot temperature | * Send data to database * Display data (non-blinded GUI and locked admin screen) |
| Start time | * Send treatment start time to App |  | * Send start time to database |
| End time | * Send treatmetn end time to App |  | * Send end time to database |
| Hot cycle # duration | * Measure actual time for each hot cycle and send data to App | * Receive command to continue to heat | * Send |
| Current cycle time remaining | * Timer to control cycle duration * Send control board command to continue current cycle | * Receive command to continue hot or cold cycle * continue to heat or cool | * Display time remaining |
| Hot/cold cycle switch | * Send control board command to switch cycles * Update current cycle time remaining to new cycle duration * Save timestamp to calculate cycle duration (end of previous cycle and start of next cycle) | * Receive command to switch cycles * Transitions between cycles | * Display new cycle screen and time remaining * Send data to database |
| Get Event (from existing board) | * Receive command from App if sent * Send command to board * Receive data from control board * Send data to App | * Receive command from PMC * Send data to PMC | * Send command to PMC * Receive data from PMC * Send data to database |
| Get Event (from PMC) | * Receive command from App |  |  |
| Error codes | * Mitigate risks caused by errors when possible * Send data to App | * Send error codes to PMC * Receive commands from PMC | * Display errors or warnings as appropriate * Send data to database |
| New firmware files | * Receive firmware from app * Update firmware when appropriate * Send confirmation new firmware was installed to App | * Receive new firmware from PMC (nice to have) | * Send firmware to PMC * Send command to PMC to update firmware * Receive confirmation * Send data to database |