主状態→ イベント↓	NULL	初期化	監視中	燃焼	消火	エラー	待機
HAOS起動 svev_startup	[svact_initialize_2] 全カウンタをクリア(clear_all_counter) 主状態→「初期化」 副状態→「消火」 待機遷移指示→false 補足→null センサー取込タイマー開始（0秒開始） if sv_enable_autotimer == true{ script.set_autostart } sv_sts_keep→null	ー	ー	ー	ー	ー	ー
センサー取込タイマー （svtm_pull_value）満了 svev_pull_value	ー	[-] 値取込サブルーチン 値取込結果処理サブルーチン(引数=待機){ if input_boolean.sv_sts_sensor == false { エラー連続カウンタ ++ if エラー連続カウンタ ≧ センサーエラー継続数 { 主状態→「エラー」 補足→「センサー取得エラー」 ブザー鳴音タイマー開始（0秒開始） } else { センサー取込タイマー開始 } } else { エラー連続カウンタクリア センサー取込タイマー開始 主状態→(引数1) } } 値取込サブルーチン : script.save_sensor_values 値取込結果処理サブルーチン : script.check_sensor_status	[-] 値取込サブルーチン 値取込結果処理サブルーチン(引数=sv_sts)	（同左）	（同左）	ー	（同左）
監視タイマー（svtm_monitor）満了 svev_monitor	ー	ー	[svact_monitor] case 現温度 < 下限、かつ副状態が「消火」{ PowerOnリトライカウンタをクリア 燃焼リトライカウンタをクリア Plug(root)をON PowerOn開始タイマー開始（0秒開始） 主状態→「燃焼」 } case（現温度 ＞上限、かつ副状態が「燃焼」）or 　　（ 待機遷移指示 == true ）{ 消火サブルーチン(script.start_putout) { 消火リトライカウンタをクリア 消火タイマー開始（0秒開始） サーボで押しボタンをPUSH 主状態→「消火」 } } default { 監視タイマー開始 主状態→（維持） }	ー	ー	ー	ー
PowerOn監視タイマー （svtm_mon_poweron）満了 svev_mon_poweron	ー	ー	[-] ログ（警告）	[svact_mon_poweron] PowerOnリトライカウンタ ++ case Plug(root)がON { 燃焼タイマー開始（0秒開始） サーボで押しボタンをPUSH } case PowerOnリトライカウンタ ＞ PowerOnリトライリミット { 主状態→「エラー」 補足→「PowerOnエラー」 ブザー鳴音タイマー開始（0秒開始） } default { Plug(root)をON PowerOn監視タイマー開始 }	ー	ー	ー
燃焼タイマー満了（svtm_burn） svev_burn	ー	ー	[-] ログ（警告）	[svact_burn] 燃焼継続カウンタをクリア 燃焼監視タイマー開始	ー	ー	ー
燃焼監視タイマー （svtm_mon_burn）満了 svev_mon_burn	ー	ー	ー	[svact_mon_burn] if Plug(root)の電力 ≧ 燃焼中電力値 { 燃焼継続カウンタ ++ ; if 燃焼継続カウンタ ≧ 継続数 { if 待機遷移指示 == true { 消火サブルーチン } else { 主状態→「監視中」 副状態→「燃焼」 監視タイマー開始（0秒開始） } } else { 燃焼監視タイマー開始 } } else { 燃焼継続カウンタクリア 燃焼リトライカウンタ ++ if 燃焼リトライカウンタ ＞ 燃焼リトライリミット { 主状態→「エラー」 補足→「燃焼確認エラー」 ブザー鳴音タイマー開始（0秒開始） } else { 燃焼タイマー開始 } }	ー	ー	ー
消火タイマー（svtm_putout）満了 svev_putout	ー	ー	ー	ー	[svact_putout] 消火継続カウンタをクリア 消火監視タイマー開始	ー	ー
消火監視タイマー （svtm_mon_putout）満了 svev_mon_putout	ー	ー	ー	ー	[svact_mon_putout] if Plug(root)の電力 ＜ 燃焼中電力値 { 消火継続カウンタ ++ ; if 消火継続カウンタ ≧ 消火継続数 { if 待機遷移指示 == true { 待機遷移指示→false 主状態→「待機」 副状態→「消火」 } else { 主状態→「監視中」 副状態→「消火」 監視タイマー開始（0秒開始） } } else { 消火監視タイマー開始 } } else { 消火継続カウンタクリア 消火リトライカウンタ ++ if 消火リトライカウンタ ＞ 消火リトライリミット { ブザー鳴音タイマー開始 主状態→「エラー」 補足→「消火エラー」 } else { 消火タイマー開始 } }	ー	ー
ブザー鳴音タイマー （svtm_buzzer）満了 svev_buzzer	ー	ー	ー	ー	ー	[svact_buzzer] スマホへ通知	ー
停止・再開ボタン - Press svev_press_stopcont **※event_dispatcherAsyncへ分岐**	ー	script.sw_button_feedback(false)	[save_sts] if sv_sts_keep == '' { sv_sts_keep ← sv_sts }	script.save_sts	script.save_sts	script.sw_button_feedback(false)	script.save_sts
停止・再開ボタン - Release svev_release_stopcont **※event_dispatcherへ分岐**	ー	ー	[svact_release_stopcont_1] if sv_sts_keep == sv_sts { if 副状態 ==「燃焼」{ 待機遷移指示→true 消火サブルーチン(script.start_putout) } else { 主状態→「待機」 } script.sw_button_feedback(success=true) } else{ script.sw_button_feedback(success=false) } sv_sts_keep←''	[svact_release_stopcont_2] if sv_sts_keep == sv_sts { 待機遷移指示→true script.sw_button_feedback(true) } else{ script.sw_button_feedback(false) } sv_sts_keep←''	script.svact_release_stopcont_2	ー	[svact_release_stopcont_3] if sv_sts_keep == sv_sts { [svact_startmon]{ clear_all_counter 待機遷移指示→false センサー取込タイマー開始（0秒開始） 監視タイマー開始（0秒開始） 状態→「監視中」 } script.sw_button_feedback(success=true) } else{ script.sw_button_feedback(success=false) } sv_sts_keep←''
停止・再開ボタン - GUIボタン用 svev_push_stopcont	ー	ー	[svact_push_stopcont_1] if 副状態 ==「燃焼」{ 待機遷移指示→true 消火サブルーチン(script.start_putout) } else { 主状態→「待機」 } sv_sts_keep←''"	[svact_push_stopcont_2] 待機遷移指示→true sv_sts_keep←''	script.svact_push_stopcont_2	ー	[svact_push_stopcont_3] script.svact_startmon sv_sts_keep←''
状態同期（監視中-燃焼）ボタン 手動で燃焼した場合に使用 svev_forcesync_burn	ー	ー	[sync_force] sync_force(引数1=監視中, 引数2=燃焼){ script.clear_all_counter 待機遷移指示→false センター取込タイマー開始(0秒開始) if 引数1 != 待機 { 監視タイマー開始(0秒開始) } 状態→(引数1) 副状態→(引数2) }	（同左）	（同左）	（同左）	（同左）
状態同期（監視中-消火）ボタン 手動で消火した場合に使用 svev_forcesync_putout	ー	ー	script.sync_force(引数1=監視中, 引数2=消火)	（同左）	（同左）	（同左）	（同左）
状態同期（待機）ボタン 地震等でストーブが自動停止した場合に使用 svev_forcesync_idle	ー	ー	script.sync_force(引数1=待機, 引数2=消火)	（同左）	（同左）	（同左）	（同左）

主状態→ イベント↓	NULL	初期化	監視中	燃焼	消火	エラー	待機
2時間タイマーON/OFF svev_push_2hourbtn	ー	[svact_toggle_svtm_2hour] 2時間タイマーが開始していないなら開始し、開始しているなら停止する。	(同左)	(同左)	(同左)	ー	ー
2時間タイマー（svtm_2hour）満了 svev_2hour	ー	ー	[svact_2hour] if 副状態 ==「燃焼」{ 待機遷移指示→true 消火サブルーチン(script.start_putout) } else { 主状態→「待機」 }	待機遷移指示→true	待機遷移指示→true	ー	ー
自動起動ON/OFF svev_autostart_onoff	ー	[svact_autostart_onoff] if sv_enable_autotime == on { [set_autostart]{ 次回タイマー日時(sv_next_autostart)を設定する ┗当日のsv_autotime前なら、当日のsv_autotime ┗当日のsv_autotime以降なら、翌日のsv_autotime 自動起動タイマー(svtm_autostart)を開始する ┗次回タイマー日時 － 現在日時 が満了時刻 } } else{ 自動起動タイマ(svtm_autostart)を解除する。 }	(同左)	(同左)	(同左)	(同左)	(同左)
自動起動タイマー （svtm_autostart）満了 svev_autostart	ー	script.set_autostart	script.set_autostart	[svact_autostart_through] if 待機遷移指示 == true { 待機遷移指示→false } script.set_autostart	script.svact_autostart_through	script.set_autostart	script.set_autostart script.svact_startmon
自動起動時間変更(変更後5秒後にトリガー) svev_change_autostart	ー	[svact_change_autostart] if sv_enable_autotime == on { script.set_autostart }	(同左)	(同左)	(同左)	(同左)	(同左)