

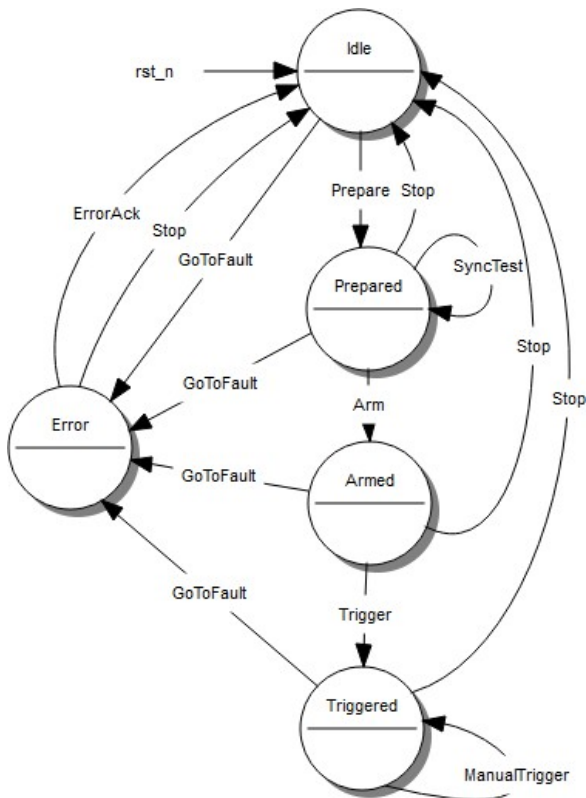


Descripción del control láser

- El control del láser se centra en una máquina de estados finitos (FSM) controlada por eventos.
- La FSM es una aplicación que se ejecuta en la placa LazServ integrada en el sistema láser.
- La propia FSM recibe órdenes de comandos remotos a través de LAN o interfaz RS232 (no cubierto en este manual).

Máquina de estados

- Representación de la máquina de estados:



- Para describir el comportamiento de la FSM se utilizan estados y transiciones:
- Estados:
 - Sólo se permite un estado en cada momento.
 - El FSM puede cambiar de un estado a otro en respuesta a un comando remoto.
 - El estado en sí es una variable, que se puede leer en cualquier momento, pero no se puede escribir.
- Transiciones
 - Sólo se permite una transición en un momento dado.
 - Los eventos se envían a una cola mientras el FSM está en transición.
 - Una transición exitosa llevará al estado planificado, una fallida puede dejar el estado sin cambios o generar el evento GoToFault.
 - La transición se compone de una secuencia lineal de comandos escritos en un lenguaje de script sencillo. Las secuencias se enumeran en la tabla SECUENCIAS.
 - La duración de la transición no está limitada, una aplicación externa puede sondear el estado actual del FSM.

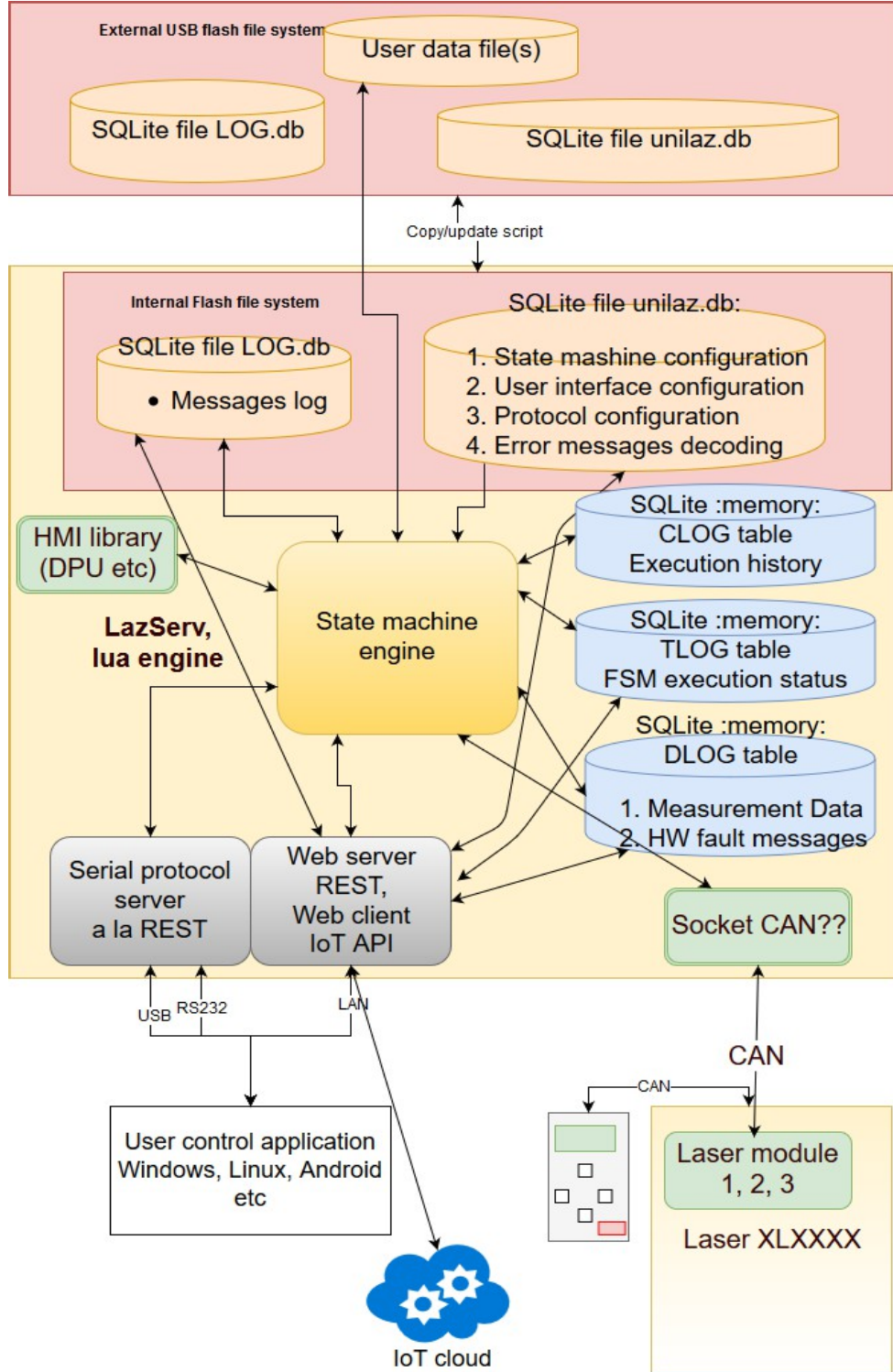
Implementación de la máquina de estados

- La máquina de estados se implementa como una aplicación ejecutable de Linux.
- La aplicación se encarga del control y la comunicación del FSM.
- La aplicación FSM está escrita en LUA.

Servicios de comunicación

- Para gestionar la comunicación entre la FSM y el controlador remoto, se ejecuta un servidor HTTP en el puerto 8081 de la placa de control.
- La API REST sobre TCP/IP permite controlar el FSM de forma remota.
- Los comandos y respuestas REST pueden introducirse/visualizarse mediante un navegador normal.
- Además del servidor REST en el PC de control, se ejecutan servidores adicionales en los puertos 80 y 8080 del controlador de comunicación. El servidor http en 80

- El servidor http en 8080 soporta REST API y se utiliza para la comunicación directa con los módulos individuales del sistema. Consulte el documento "Descripción del protocolo API REST sobre LAN o WLAN" para más detalles.
- Los servidores 80, 8080 utilizan protocolos diferentes que no se tratan en este manual.
- El FSM se comunica con el hardware del láser a través de un puerto COM, virtual o auténtico, o bien a través de LAN/WLAN.



Bloques esenciales

1. Motor de la máquina de estados. Se emplea un ciclo infinito: recibir comandos - procesar eventos - ejecutar secuencias - publicar resultados.
 2. Servidor web/REST. Procesa las solicitudes REST GET y compone los datos de retorno. Toda la comunicación entre la aplicación de usuario y el FSM láser pasa por este servidor.
 3. Base de datos de configuración, archivo SQLite unilaz.db.
 4. Base de datos de registro de sesiones, archivo SQLite log.db. Registra los mensajes de error en el modelo normal y todos los pasos en el modo de depuración.
 5. Base de datos de estado de ejecución, archivo de memoria SQLite, tabla TLOG. Contiene actualizaciones sobre el estado de ejecución de las secuencias.
 6. Base de datos de datos, archivo de memoria SQLite, tabla DLOG.
- Los archivos SQLite son accesibles mediante consultas especializadas.

Peticiones de interfaz remoto/usuario:

- El FSM comienza ejecutando la secuencia **Init**. Una vez finalizada la secuencia de **inicio**, el FSM entra en modo de espera de eventos.
- Los eventos son generados por comandos remotos.
- URL de solicitud/comando: `protocol://host/REST/HTTP_CMD/query`
 - Tanto las peticiones como los comandos utilizan el protocolo http GET, por lo que el navegador de Internet es suficiente para probar la comunicación y el control.
 - **protocolo**: http
 - **host**: xxx.xxx.xxx.xxx:8081.
 - **REST/HTTP_CMD** - cadena de identificación del FSM
 - **query** - solicitud de comando o parámetro

formato de la consulta

- **query** comienza con el símbolo de interrogación. En la descripción siguiente se omite '?'
- La palabra '?' va seguida de la palabra query o comando y de los parámetros separados por el símbolo '/'.
- All:
 - **query** - solicitud de estado de ejecución, variables o datos de sensores. La consulta no cambia el estado del FSM.
 - **comando** - instrucciones para el FSM que alterarán el estado.
- Uso de comandos/consultas:

Consulta LIST

- Devuelve los resultados de la consulta SELECT de los archivos de base de datos SQLite unilaz.db y log.db.
- El uso más obvio del comando **LIST** es leer el registro de mensajes de ejecución o leer la lista de mensajes codificados.
- Uso del comando LIST:
 - `LIST/XXXX[/YYYY]`
 - Esto resultará en una consulta a la base de datos: **SELECT YYYY FROM XXXX**
 - YYYY son nombres de columnas separados por comas. YYYY se sustituirá por * en caso de que no se proporcione ningún parámetro, lo que dará como resultado todas las columnas de la lista. Se puede utilizar la sentencia DISTINCT.
- XXXX debe empezar por el nombre de la tabla, pero no se limita a él. Se pueden utilizar las sentencias GROUP BY, ORDER BY y LIMIT. Consulte el manual del lenguaje SQLite para más detalles https://sqlite.org/lang_select.html.
- Nombres de tabla:
 - **SEQUENCES** - secuencias de comandos FSM
 - **MSG** - tabla de decodificación de mensajes codificados, campos:
 - **ERROR** tipo INTEGER - código de mensaje que la aplicación responde a todas las consultas y comandos.
 - **ID** type INTEGER - código adicional, utilizado para decodificar fallos de hardware.
 - **FUNCTION** tipo STRING, proceso, propietario de este mensaje, nombre.
 - **FSTRING** tipo TEXT, cuerpo del mensaje.
 - **COMMENT** type TEXT, información adicional, por ejemplo, qué hacer a continuación.
 - **CLOG** - registro de ejecución de la sesión.
 - **TIME** tipo REAL, hora en que se registró el mensaje en segundos transcurridos desde, 1900 01 01
 - **STEP** tipo INTEGER, SECUENCIAS
 - **FAULT** tipo INTEGER,
 - **RESULT** tipo TEXT,
 - **SRC** tipo TEXT
 - **COM** - cadenas de formato de respuesta del protocolo.
- ejemplos de consulta:
 - `http://192.168.0.160:8081/REST/HTTP_CMD/?LIST/MSG` - lista de mensajes codificados
 - `http://192.168.0.160:8081/REST/HTTP_CMD/?LIST/SEQUENCES/DISTINCT SECUENCIA` para ver todos los comandos FSM disponibles.
 - `http://192.168.0.160:8081/REST/HTTP_CMD/?LIST/CLOG` para obtener el registro de ejecución
 - `http://192.168.0.160:8081/REST/HTTP_CMD/?LIST/CLOG WHERE TIME > 3574141740.945` para obtener el registro de ejecución desde un momento determinado.

momento.




Consulta DATA.

- Devuelve los resultados de la consulta SELECT de la tabla de datos SQLite. La tabla de datos contiene las lecturas de los sensores e informa sobre los cambios en los valores de los registros. Los informes sobre cambios en el registro de 'Código de error' se publican en esta tabla.
- La tabla de datos está organizada en tres columnas:
 - **TIME** - fecha y hora de los datos. Se trata de lecturas del temporizador local, número entero. Una cuenta corresponde a 1us. La marca de tiempo depende del sistema, sin valor absoluto significativo.
 - **CHAN** - canal de registro. Los datos de diferentes fuentes se organizan en canales separados. El número de canal se asigna durante la inicialización mediante el comando **logstart**, véase la columna VALUE de la tabla SEQUENCES.

IND	SEQUENCE	COMMAND	ADDRESS	REGISTER	VALUE
11	Init	logstart	SY3PL50M:32	Error Code	1
12	Init	logstart	LDD1A:18	Error Code	2
13	Init	logstart	PHD1K000:3	Data	3
14	Init	logstart	PHD1K000:4	Data	4

- **DATA** - lecturas actuales en formato de doble flotador.
- El formato del comando es DATA/ChannelNo[/FromTime].
- Ejemplos de consulta:
 - http://192.168.0.160:8081/REST/HTTP_CMD/?DATA/2 - obtener todos los datos del canal No 2
 - http://192.168.0.160:8081/REST/HTTP_CMD/?DATA/2/145886649649 - datos del canal n° 2, donde HORA> 145886649
- La tabla de datos mantiene 5000 registros en total. A la máxima velocidad de registro, por ejemplo, 5 sensores x 1000Hz, el buffer de frecuencia mantiene los datos con una antigüedad máxima de un segundo. Los datos deben ser consultados más a menudo que la edad máxima de los datos para mantener la consistencia.
- El software de aplicación que necesita un conjunto de datos consistente del canal debe utilizar la consulta DATA/ChannelNo al principio. Posteriormente, **la consulta DATA/ChannelNo /FromTime**. FromTime es la marca de tiempo del último registro de datos recibido.

Comando EXE

- Coloca el comando FSM en la cola de ejecución.
- En el momento de la contabilización en la cola de mando, se asigna el ticket. El ticket sirve para la recepción en tiempo real de la orden. El ticket se utiliza posteriormente para recuperar el estado de ejecución de la orden.
- El estado real de la orden no se conoce en el momento del envío, por lo que una respuesta positiva significa el éxito del envío, pero no de la ejecución en sí.
- Véase la consulta **CES** sobre la obtención del estado de ejecución.
- El formato del comando es EXE/Comando[/Parámetro].
- Ejemplos de comandos:
 - http://192.168.0.160:8081/REST/HTTP_CMD/?EXE/Stop 
 - http://192.168.0.160:8081/REST/HTTP_CMD/?EXE/Fire 
 - http://192.168.0.160:8081/REST/HTTP_CMD/?EXE/Fire 
- Los comandos y parámetros disponibles figuran en la documentación del producto.

Consulta RDVAR

- Devuelve el valor de la variable de proceso.
- Las variables de proceso se utilizan para pasar parámetros entre los pasos de la secuencia.
- Las variables son de sólo lectura desde el punto de vista del control remoto.
- Hay varias variables predefinidas:
 - State' - estado actual del FSM
 - LogBlab' - extensión del registro. 0 - sólo se registran los errores, 2 - se registran todos los pasos de la secuencia.
 - x' - variable utilizada para pasar el parámetro del comando. Por ejemplo, el comando EXE/Amplification/50 hace lo siguiente: establece la variable x en 50 y ejecuta la secuencia "Amplification".

secuencia "Amplificación".

- ProductID' - tipo de producto
- ProductSN' - número de serie
- Ejemplo de consulta RDVAR:
 - http://192.168.0.160:8081/REST/HTTP_CMD/?RDVAR/State - devuelve el estado actual.
 - http://192.168.0.160:8081/REST/HTTP_CMD/?RDVAR/LogBlab - devuelve el alcance del registro.

Consulta CES.

- Devuelve el estado del comando que se está ejecutando o ha finalizado.
- El formato del comando es CES[/Timestamp].
- Timestamp es el marcador del comando y se recibe en la respuesta del comando EXE.

- Ejemplo de comando:
 - http://192.168.0.160:8081/REST/HTTP_CMD?CES/3574677021.391928
- El comando sin **marca de tiempo** devuelve el estado del último comando recibido de la cola.
- La consulta CES devuelve el código de estado de la consulta y los siguientes parámetros de ejecución del comando:
 - Código de estado de ejecución del comando:
 - 3 - el comando se envía a la cola
 - 2 - comando recibido de la cola por el FSM
 - 1 - la orden se está ejecutando
 - 0 - la orden se ha ejecutado correctamente
 - >0 - la orden finaliza con error
 - Índice de paso (IND). Índice (valor de la columna IND) del paso actual en ejecución o del último paso para el comando finalizado. Conocer el valor IND permite entender qué está haciendo ahora el FSM.
 - Resultado. Resultado de la ejecución del paso de la secuencia. En ciclos de espera largos, ver el resultado intermedio puede ayudar a comprender el proceso en sí. Por ejemplo, si el sistema espera a que la temperatura del calentador suba por encima de algún umbral, puede ser útil ver la temperatura actual. El resultado guarda información importante sobre las consecuencias de un error en la ejecución de la secuencia. Las palabras clave que explican las consecuencias que siguen a la cadena "Next:" pueden ser:
 - GoToFault** - la ejecución de la secuencia finaliza en este punto y el comando 'GoToFault' es enviado a la cola. En general, el resultado será algún estado indicador de fallo. Tenga en cuenta, GoToFault será el nuevo comando y el antiguo marcado no sho
 - Saltarse el resto** - la ejecución de la secuencia finaliza en este punto, los pasos restantes de la secuencia se saltan. En general, el resultado será que no habrá cambios de estado. Los guardias generan este error.
 - Ignorando error** - no habrá consecuencias, excepto que el último paso fue erróneo.
 - Fuente. HTTP_CMD.vi para el comando de control remoto.
 - Marca de tiempo de la última operación

Respuestas a las consultas.

- Las respuestas a las consultas se formatean de acuerdo con la información de la columna RES_HTML en la tabla COM, archivo unilaz.db

NOMBRE_COM	FUNCIÓN	RES_PAR_COUT	RES_HTML	DESCRIPCIÓN
EXE	exeSEQM	2	%d Comprobar estado	2. Comando ticket(timestamp)
RDVAR	RetTrueValue	2	%d %s %s	2. Valor de la variable 3. Tipo
CES	checkID	6	%d %d %d %s %s %f	2. Código de estado de ejecución del comando 3. Índice de paso (IND) 4. Resultado 5.Origen 6. Fecha y hora de la última operación
LISTA	SQL	2	%d <code>%s</code> 	2. Resultado de la consulta SQL en forma de lista
DATOS	Datos SQL	2	%d <code>%s</code> 	2. Resultado de la consulta SQL en forma de lista

Allí:

- COM_NAME** - nombre de la consulta/comando tal como aparece en la consulta.
- FUNCTION** - nombre de la función interna. Debe dejarse sin cambios, porque es el enlace a la función real.
- RES_PAR_COUT** - parámetros que se pasan a la función de formato en respuesta a la consulta.
- RES_HTML** - cadena de formato que especifica cómo desea que la función convierta los argumentos de entrada en la cadena resultante.
- DESCRIPTION** - descripciones de los parámetros. El primer parámetro siempre es el código de respuesta, por lo tanto se omite.

Reglas generales:

- El primer parámetro es siempre el código de respuesta a la consulta. Un código cero significa que la consulta/comando fue aceptada y que los siguientes resultados son válidos. Un código distinto de cero significa que la consulta no puede devolver ningún resultado significativo debido al error que representa el código.
- El(los) siguiente(s) parámetro(s) es(son) el(los) resultado(s) real(es) de la consulta.
- Las respuestas a las consultas tienen un formato mínimo, sólo para mantener la legibilidad en los navegadores modernos. Cambiando las cadenas de formato existe la posibilidad de cumplir con muchos de los estándares de codificación de la información: XML, JSON, HTML.
- El formato de respuesta es sencillo para las consultas EXE, RDVAR, CES, sólo hay una cadena de formato.
- El formato de respuesta para las consultas LIST y DATA se divide en dos pasos:

1. %d
<code>%s</code>

- es el sobre general. Allí %d es código de respuesta, %s es cuerpo de respuesta.

2. El símbolo '|' separa evenlope del formato de fila.

- Las reglas de formato de fila son:

1. consulta DATA. Hay tres cadenas separadas por el símbolo '|': 1. inicio de fila (cadena vacía) 2. separador (punto y coma) 3. final de fila (salto de línea simple).
2. Consulta LIST. Hay cuatro cadenas separadas por el símbolo '|': 1. inicio de columna (cadena vacía) 2. separador de columna (punto y coma) 3. inicio de fila (cadena vacía) 3. terminador de fila (salto de línea simple).

Ejemplos de respuesta

Consulta de comandos disponibles: http://192.168.0.160:8081/REST/HTTP_CMD/?LIST/SEQUENCES/DISTINCT%SECUENCIA...



Response

```
0<br><code>SomeEvent;<br>Init;<br>SyncMode;<br>Amplification;<br>Watchdog;<br>GoToFault;<br>Stop;<br>Fire;<br>EnMode;<br></code>
```

Consulta de datos del canal 2: http://192.168.0.160:8081/REST/HTTP_CMD/?DATA/2...



Respuesta

```
0<br><code>1631885132;0.000000;<br>1631885509;0.000000;<br>1631885881;0.000000;<br>1631886255;0.000000;<br>1631886631;0.000000;<br>
```

Comando 'Stop': http://192.168.0.160:8081/REST/HTTP_CMD/?EXEC/Stop...

Respuesta

```
0<br><a href="?CES/3575601570403">Comprobar estado</a>
```

Consulta del estado de ejecución del comando 'Disparar' con el ticket:

http://192.168.0.160:8081/REST/HTTP_CMD/?CES/3574823433.061258...



Respuesta, comando ejecutado con éxito:

```
0<br>0<br>142<br>%22Idle%22 <br>HTTP_CMD.vi <br>09:39:30.694 2017.04.21
```

Respuesta, error sucedido:

```
0<br>310<br>79<br>Next:%20Skipping%20rest%20 <br>HTTP_CMD.vi <br>09:42:21.460 2017.04.21
```

Consulta de serie del producto: http://192.168.0.160:8081/REST/HTTP_CMD/?REVAR/ProductSN...

Respuesta:

```
0<br>"001"
```

Tratamiento de errores y sistema de información

El FSM gestiona los errores procedentes de las siguientes fuentes:

1. Módulo de ejecución FSM
2. Red CAN
3. Servidor http
4. Módulos de hardware

- Cada paso de la secuencia tiene asignado un tipo de gestor de errores. En caso de que el objetivo del paso no se pueda alcanzar, el controlador de errores realiza algunas acciones predefinidas.

Tipos de gestores de errores

Nombre	Parámetro	Acción(es)	Uso típico
ResetErr	-	Se borra el error, se publica 'Clean completion' como resultado en el log con el código de éxito	Se utiliza junto con el comando waitfor que actúa como retardo de tiempo fijo
IgnoreErr	-	El error se borra, 'Siguiente: Ignorar error' se envía como resultado al registro con el código de error	Útil durante la depuración para no interrumpir la ejecución pero ver el código de error en el registro
SkipRestOnErr	Opcional, sustituye el código	Siguiente: 'Skipping rest' se publica como resultado en el log con el código de error original o si se suministra con código sustituto. Se termina la ejecución de la secuencia	Se utiliza en las guardias al principio de la secuencia. En caso de que no se superen las guardias, no se ejecutan más pasos y el estado del FSM no cambia.
FaultOnErr	Opcional, código de sustitución	El evento 'GoToFault' se envía a la cola, 'Next: GoToFault' se publica como resultado en el registro con el código de error original o si se suministra con código sustitutivo. La secuencia actual finaliza, la secuencia GoToFault se recupera de la cola y se ejecuta.	Se utiliza para dirigir el FSM al modo de fallo

- El código de sustitución en SkipRestOnErr y FaultOnErr se utiliza para sustituir el mensaje de error estándar por información relacionada con el sistema concreto. Por ejemplo, en caso de que se compruebe en la secuencia algún bit que represente el estado de enclavamiento de seguridad, el FSM publicará normalmente el error de que este bit tiene el valor x en lugar del requerido y. Esto hace que la resolución de problemas sea sofisticada.