

Test ID	Tipo	Nombre del Test	Descripción
ADQ_001	Positivo	Identificar cada ds18b20 conectado al bus 1-wire	Identificar el ID único de cada sensor detectado en el bus 1-wire
ADQ_002	Positivo	Detectar presencia de cada ds18b20 identificado	Detectar la presencia de cada sensores de temperatura conectado al bus 1-wire
ADQ_003	Positivo	Adquirir valores de temperatura	Adquirir un valor de temperatura por cada sensor conectado al bus 1-wire
ADQ_004	Positivo	Adquirir periódicamente temperatura	Adquirir periódicamente un valor de temperatura de cada sensor conectado al bus 1-wire.
ADQ_005	Positivo	Leer configuración de ds18b20	Leer el contenido de la memoria interna de cada sensor de temperatura conectado al bus 1-wire
ADQ_006	Positivo	Escribir configuración de ds18b20	Escribir en la memoria interna de cada sensor de temperatura conectado al bus 1-wire
ADQ_007	Positivo	Convertir temperatura a celcius	Convertir el valor leído de cada sensor de temperatura de complemento a 2 a decimal.
ADQ_008	Positivo	Leer el estado del controlador de bus 1-wire	Leer en qué estado se encuentra el controlador del bus 1-wire
ADQ_009	Positivo	Desactivar el controlador del Bus 1-wire	Pasivar el controlador de bus 1-wire
ADQ_010	Positivo	Activar el controlador del Bus 1-wire	Activar el controlador de bus 1-wire
ADQ_011	Negativo	No identificar los ds18b20 conectados al bus 1-wire	Test ADQ_001 con inyección de falla mediante un función mock
ADQ_012	Negativo	No detectar presencia de ds18b20 identificados	Test ADQ_002 con inyección de falla mediante un función mock
ADQ_013	Negativo	No adquirir valores de temperatura	Test ADQ_003 con inyección de falla mediante un función mock
ADQ_014	Negativo	No adquirir periódicamente temperatura	Test ADQ_004 con inyección de falla mediante un función mock
ADQ_015	Negativo	No leer configuración de ds18b20	Test ADQ_005 con inyección de falla mediante un función mock
ADQ_016	Negativo	No escribir configuración de ds18b20	Test ADQ_006 con inyección de falla mediante un función mock
ADQ_017	Negativo	No convertir temperatura a celcius	Test ADQ_007 con inyección de falla mediante un función mock
ADQ_018	Negativo	No leer el estado del controlador de bus 1-wire	Test ADQ_008 con inyección de falla mediante un función mock
ADQ_019	Negativo	No desactivar el controlador del Bus 1-wire	Test ADQ_009 con inyección de falla mediante un función mock
ADQ_020	Negativo	No activar el controlador del Bus 1-wire	Test ADQ_010 con inyección de falla mediante un función mock
ADQ_021	Rango	Rango válido de valor de temperatura	Verificar el rango de temperatura obtenido en el test ADQ_003
ADQ_022	Rango	Rango válido de período de adquisición	Verificar que se acepta un valor de período con rango válido en el test ADQ_004
ADQ_023	Rango	Rango válido de valores de configuración	Verificar que se aceptan valores de configuración con rango válido en el test ADQ_006
ADQ_024	Rango	Rango válido de temperatura en Celcius	Verificar que el valor de temperatura convertido a grados Celsius esté en el rango válido
ADQ_025	Rango	Rango válido de estado del controlador de bus	Verificar que el estado del controlador del bus 1-wire esté en el rango válido definido en ow_mef_state_t
ADQ_026	Rango	Rango inválido de valor de temperatura	Verificar que se detecta un valor de temperatura fuera de rango pasado mediante una función mock
ADQ_027	Rango	Rango inválido de período de adquisición	Verificar que se detecta un valor de período fuera de rango pasado mediante una función mock
ADQ_028	Rango	Rango inválido de valores de configuración	Verificar que se detectan valores de configuración fuera de rango pasados mediante una función mock
ADQ_029	Rango	Rango inválido de temperatura en Celcius	Verificar que se detecta un valor de temperatura fuera de rango pasado mediante una función mock
ADQ_030	Rango	Rango inválido de estado del controlador de bus	Verificar que se detecta un estado del controlador fuera de rango pasado mediante una función mock

Test ID	Tipo	Nombre del Test	Descripción
HMI_001	Positivo	Obtener la posición del cursor en la pantalla	Obtener la posición X,Y de dónde se encuentra el cursor en la pantalla de la terminal serie
HMI_002	Positivo	Desplazar el cursor al cmdline	Desplazar el cursor a la posición X,Y de la línea de comandos
HMI_003	Positivo	Desplazar el cursor a la barra de estado	Desplazar el cursor a la posición X,Y de la barra de estado
HMI_004	Positivo	Limpiar la pantalla	Enviar los comandos para borrar la pantalla y desplazar el cursor a la posición home
HMI_005	Positivo	Imprimir el encabezado	Enviar a la UART un string constante con el encabezado
HMI_006	Positivo	Imprimir menú de opciones contextual	Enviar a la UART un puntero a un string válido pasado por referencia.
HMI_007	Positivo	Imprimir mensaje de error	Imprimir un mensaje de error en la línea de comandos cuando se ingrese una opción no válida.
HMI_008	Positivo	Imprimir un mensaje de estado	Imprimir un mensaje de estado en la barra de estado
HMI_009	Positivo	Estado inicial de la MEF menú	Verificar el estado inicial de la MEF que maneja menú
HMI_010	Positivo	Obtener el estado de la MEF menú	Obtener el estado actual de la MEF que maneja el menú
HMI_011	Negativo	No obtener la posición del cursor en la pantalla	Test ADQ_001 con inyección de falla mediante un función mock
HMI_012	Negativo	No desplazar el cursor al cmdline	Test ADQ_002 con inyección de falla mediante un función mock
HMI_013	Negativo	No desplazar el cursor a la barra de estado	Test ADQ_003 con inyección de falla mediante un función mock
HMI_014	Negativo	No limpiar la pantalla	Test ADQ_004 con inyección de falla mediante un función mock
HMI_015	Negativo	No imprimir el encabezado inválido	Test ADQ_005 con inyección de falla mediante un función mock
HMI_016	Negativo	No imprimir menú de opciones contextual inválido	Test ADQ_006 con inyección de falla mediante un función mock
HMI_017	Negativo	No imprimir mensaje de error inválido	Test ADQ_007 con inyección de falla mediante un función mock
HMI_018	Negativo	No imprimir un mensaje de estado inválido	Test ADQ_008 con inyección de falla mediante un función mock
HMI_019	Negativo	Estado inicial de la MEF menú inválido	Test ADQ_009 con inyección de falla mediante un función mock
HMI_020	Negativo	No obtener el estado de la MEF menú	Test ADQ_010 con inyección de falla mediante un función mock
HMI_021	Rango	Rango válido para posición X del cursor	Verificar que se acepta un valor de posición X con rango válido en la función que desplaza el cursor
HMI_022	Rango	Rango válido para posición Y del cursor	Verificar que se acepta un valor de posición Y con rango válido en la función que desplaza el cursor
HMI_023	Rango	Rango válido para estado de MEF menú	Verificar que el estado de la MEF menú esté en el rango válido definido en mef_menu_state_t
HMI_024	Rango	Rango inválido para posición X del cursor	Verificar que se detecta un valor de posición X del cursor fuera de rango pasado mediante una función mock
HMI_025	Rango	Rango inválido para posición Y del cursor	Verificar que se detecta un valor de posición Y del cursor fuera de rango pasado mediante una función mock
HMI_026	Rango	Rango inválido para estado de MEF menú	Verificar que se detecta un estado de la MEF menú fuera de rango

Test ID	Tipo	Nombre del Test	Descripción
ALM_001	Positivo	Inicializar el FileSystem	Verificar la correcta inicialización del sistema de archivos
ALM_002	Positivo	Listar archivos presentes en la memoria SD	Verificar que se pueden listar correctamente los archivos presentes en la SD
ALM_003	Positivo	Leer estado vacío de la memoria SD	Verificar que se se obtiene estado vacío cuando no hay archivos presentes
ALM_004	Positivo	Leer estado lleno de la memoria SD	Verificar que se se obtiene estado lleno cuando no hay lugar para más archivos en la SD
ALM_005	Positivo	Abrir un archivo en la memoria SD	Verificar que se obtiene un puntero válido al archivo en la operación f_open()
ALM_006	Positivo	Leer un archivo en la memoria SD	Verificar que se lee correctamente el contenido de un archivo de texto plano
ALM_007	Positivo	Escribir un archivo en la memoria SD	Verificar que se puede escribir en un archivo en la SD
ALM_008	Positivo	Leer el estado del módulo de almacenamiento	Leer en qué estado se encuentra el controlador de almacenamiento
ALM_009	Positivo	Desactivar el módulo de almacenamiento	Pasivar el controlador de almacenamiento
ALM_010	Positivo	Activar el módulo de almacenamiento	Activar el controlador de almacenamiento
ALM_011	Negativo	No inicializar el FileSystem	Test ADQ_001 con inyección de falla mediante un función mock
ALM_012	Negativo	No listar archivos presentes en la memoria SD	Test ADQ_002 con inyección de falla mediante un función mock
ALM_013	Negativo	No leer estado vacío de la memoria SD	Test ADQ_003 con inyección de falla mediante un función mock
ALM_014	Negativo	No leer estado lleno de la memoria SD	Test ADQ_004 con inyección de falla mediante un función mock
ALM_015	Negativo	No abrir un archivo en la memoria SD	Test ADQ_005 con inyección de falla mediante un función mock
ALM_016	Negativo	No leer un archivo en la memoria SD	Test ADQ_006 con inyección de falla mediante un función mock
ALM_017	Negativo	No escribir un archivo en la memoria SD	Test ADQ_007 con inyección de falla mediante un función mock
ALM_018	Negativo	No leer el estado del módulo de almacenamiento	Test ADQ_008 con inyección de falla mediante un función mock
ALM_019	Negativo	No desactivar el módulo de almacenamiento	Test ADQ_009 con inyección de falla mediante un función mock
ALM_020	Negativo	No activar el módulo de almacenamiento	Test ADQ_010 con inyección de falla mediante un función mock

Test ID	Tipo	Nombre del Test	Descripción
CON_001	Positivo	Obtener el estado de un módulo	Obtener un valor válido de estado cuando pide el estado de un módulo válido
CON_002	Positivo	Construir un mensaje	Construir exitosamente un mensaje con todos los campos definidos para ipcex_msg_t
CON_003	Positivo	Enviar un mensaje	Insertar exitosamente un mensaje en una cola de mensajes
CON_004	Positivo	Registrar todos los módulos	Registrar exitosamente todos los módulos en un vector de módulos
CON_005	Positivo	Enviar una señal a todos los módulos	Encolar exitosamente un mismo mensaje para todos los módulos registrados en el vector de módulos
CON_006	Positivo	Obtener exitosamente el estado de la cola de mensajes vacía	Obteter el estado correcto cuando se consulta una cola de mensajes que no tiene ningún elemento
CON_007	Positivo	Obtener exitosamente el estado de la cola de mensajes llena	Obteter el estado correcto cuando se consulta una cola de mensajes que se encuentra llena
CON_008	Positivo	Registrar todos los módulos	Registrar exitosamente todos los módulos en un vector de módulos
CON_009	Positivo	Enviar una señal a todos los módulos	Encolar exitosamente un mismo mensaje para todos los módulos registrados en el vector de módulos
CON_010	Positivo	Obtener el handler de un módulo válido	Obtener un puntero válido al manejador de eventos de un módulo cuando se requiere despachar un evento
CON_011	Negativo	No obtener el estado de un módulo	Test ADQ_001 con inyección de falla mediante un función mock
CON_012	Negativo	No lograr construir un mensaje	Test ADQ_002 con inyección de falla mediante un función mock
CON_013	Negativo	No lograr enviar un mensaje	Test ADQ_003 con inyección de falla mediante un función mock
CON_014	Negativo	No lograr registrar todos los módulos	Test ADQ_004 con inyección de falla mediante un función mock
CON_015	Negativo	No lograr enviar una señal a todos los módulos	Test ADQ_005 con inyección de falla mediante un función mock
CON_016	Negativo	No obtener exitosamente el estado de la cola de mensajes vacía	Test ADQ_006 con inyección de falla mediante un función mock
CON_017	Negativo	No obtener exitosamente el estado de la cola de mensajes llena	Test ADQ_007 con inyección de falla mediante un función mock
CON_018	Negativo	No lograr registrar todos los módulos	Test ADQ_008 con inyección de falla mediante un función mock
CON_019	Negativo	No lograr enviar una señal a todos los módulos	Test ADQ_009 con inyección de falla mediante un función mock
CON_020	Negativo	No lograr obtener el handler de un módulo válido	Test ADQ_010 con inyección de falla mediante un función mock
CON_021	Rango	Rango válido para construir mensaje	Rango válido para todos los campos pasados como parámetros para construir un mensaje
CON_022	Rango	Rango válido para enviar mensaje	Rango válido para todos los campos pasados como parámetros para enviar un mensaje
CON_023	Rango	Rango válido de señal broadcast	Rango válido para la señal que se desea enviar a todos los módulos
CON_024	Rango	Rango inválido para construir mensaje	Detectar rango inválido en los campos pasados como parámetros para construir un mensaje
CON_025	Rango	Rango inválido para enviar mensaje	Detectar rango inválido en los campos pasados como parámetros para enviar un mensaje
CON_026	Rango	Rango inválido de señal broadcast	Detectar rango inválido en la señal que se desea enviar a todos los módulos

Nombre del módulo	ADQUISICIÓN	Test diseñado por:	Patricio Bos	Test ejecutado por:	Patricio Bos	Pre con
Revisión del código	1.0	Fecha de diseño:		Fecha de ejecución:		

Test ID#	Nombre de Test	Resumen del Test	Pasos del test
ADQ_001	Identificar cada ds18b20 conectado al bus 1-wire	Identificar el ID único de cada sensor detectado en el bus 1-wire	Ejecutar algoritmo de discovery Para cada ID de dispositivo: Reset + Match ROM Enviar ID Para cada ID de dispositivo: Reset + March ROM Enviar cmd Convert T
ADQ_002	Detectar presencia de cada ds18b20	Detectar la presencia de cada sensores de temperatura conectado al bus 1-wire	Leer memoria del dispositivo Realizar en forma periódica el test ID# ADQ_003 Para cada ID de dispositivo: Reset + March ROM Enviar cmd read scratchpad Leer memoria del dispositivo
ADQ_003	Adquirir valores de temperatura	Adquirir un valor de temperatura por cada sensor conectado al bus 1-wire	Reset + March ROM Enviar cmd Convert T
ADQ_004	Adquirir periódicamente valores de temperatura	Adquirir periódicamente un valor de temperatura de cada sensor conectado al bus 1-wire.	Enviar cmd read scratchpad Leer memoria del dispositivo
ADQ_005	Leer configuración de ds18b20	Leer el contenido de la memoria interna de cada sensor de temperatura conectado al bus 1-wire	Reset + March ROM Enviar cmd write scratchpad Escribir memoria del dispositivo
ADQ_006	Escribir configuración de ds18b20	Escribir en la memoria interna de cada sensor de temperatura conectado al bus 1-wire	Reset + March ROM Enviar cmd read scratchpad Leer memoria del dispositivo
ADQ_007	Convertir temperatura a celcius	Convertir el valor leído de cada sensor de temperatura de complemento a 2 a decimal.	Ejecutar función de conversión pasándole como parámetro un valor de temperatura en formato complemento a 2
ADQ_008	Leer el estado del controlador de bus 1-wire	Leer en qué estado se encuentra el controlador del bus 1-wire	
ADQ_009	Desactivar el controlador del Bus 1-wire	Pasivar el controlador de bus 1-wire	
ADQ_010	Activar el controlador del Bus 1-wire	Activar el controlador de bus 1-wire	
ADQ_011	No detectar presencia de ds18b20		
ADQ_012	No identificar los ds18b20 detectados		
ADQ_013	No adquirir valores de temperatura		
ADQ_014	No adquirir periódicamente temperatura		
ADQ_015	No leer configuración de ds18b20		
ADQ_016	No escribir configuración de ds18b20		
ADQ_017	No convertir temperatura a celcius		
ADQ_018	No leer el estado del controlador de bus 1-wire		
ADQ_019	No desactivar el controlador del Bus 1-wire		
ADQ_020	No activar el controlador del Bus 1-wire		
ADQ_021	Rango de valor de temperatura		
ADQ_022	Rango de período de adquisición		
ADQ_023	Rango de valores de configuración		
ADQ_024	Rango de temperatura en Celcius		
ADQ_025	Rango de estado del controlador de bus 1-wire		

diciones:	Al menos 1 sensor de temperatura DS18B20 conectado.

Resultado Esperado	Resultado Obtenido	Estado
Al menos 1 ID de dispositivo válido		
Pulso de presencia del dispositivo interrogado		
Valor de temperatura		
Valores de temperatura		
8 bytes de configuración del dispositivo		
8 bytes de configuración del dispositivo iguales a los enviados		

PROYECTO:

EAMMRA

ID CASO DE PRUEBA:

NOMBRE DEL CASO:

Versión de firmware:

Test diseñado por:

Fecha de diseño:

Test ejecutado por:

Fecha de ejecución:

Pre condiciones:

Post condiciones:

ÉXITO:

FALLA:

Resumen del Test:

Step #	Descripción	Resultado Esperado	Resultado Obtenido
1			
2			
3			
4			
5			
6			
7			
8			
9			
10			
11			
12			
13			
14			
15			





PROYECTO:

EAMMRA

ID CASO DE PRUEBA:

NOMBRE DEL CASO:

Versión de firmware:

Test diseñado por:

Fecha de diseño:

Test ejecutado por:

Fecha de ejecución:

Pre condiciones:

Post condiciones:

ÉXITO:

FALLA:

Resumen del Test:

Step #	Descripción	Resultado Esperado	Resultado Obtenido
1			
2			
3			
4			
5			
6			
7			
8			
9			
10			



<b>PROYECTO:</b>		<b>EAMMRA</b>	
ID CASO DE PRUEBA:	UC01	Test diseñado por:	Patricio Bos
NOMBRE DEL CASO:	Medición de temperatura autónoma	Fecha de diseño:	01/10/18
Versión de firmware:	1.0	Test ejecutado por:	Patricio Bos
		Fecha de ejecución:	01/11/18
Pre condiciones:	2 sensores DS18B20 conectados al puerto 1-WIRE (GPIO3[0]) de la CIAA-NXP. 1 Tarjeta microSD con formato FAT32 conectada al puerto SPI de la CIAA-NXP. Terminal serie conectada en configuración 8N1 115200 a la UART-USB		
Post condiciones:	ÉXITO: Valores de temperatura y timestamp en archivo de registro en la microSD FALLA: Archivo de registro en la tarjeta de memoria microSD vacío.		
Resumen del Test:	Test de funcionamiento autónomo. Después de energizar el sistema, se inicializan los módulos y se comienza a registrar en forma periódica valores de temperatura junto con una marca de tiempo en la tarjeta de memoria microSD. Se debe esperar 1 minuto desde la inicialización del sistema para y consultar el contenido del archivo de registro desde la interfaz HMI.		
Step #	Descripción	Resultado Esperado	Resultado Obtenido
1	Energizar el sistema	Mensaje de startup en la consola	
2	(sistema) Inicialización de todos los módulos	Mensajes de inicialización de cada módulo en la consola	
3	(sistema) Expira el timeout para la tarea periódica de adquisición de temperatura. Se encola una señal sig_timeout para el módulo de adquisición	-	
4	(sistema) El módulo de adquisición recibe la señal sig_timeout y realiza una adquisición de temperatura.	-	
5	(sistema) El módulo de adquisición envía una señal sig_write al módulo de almacenamiento con el dato de temperatura medido	-	
6	(sistema) El módulo de almacenamiento recibe la señal sig_write y un valor de temperatura para guardar. El módulo de almacenamiento adquiere una marca de tiempo del RTC del sistema. El módulo de almacenamiento escribe una entrada en la tarjeta microSD con el valor de temperatura y una marca de tiempo	-	
7	Esperar que se cumpla 1 minutos desde el paso #1	-	
8	En la consola, ingresar al modo debug	Visualizar en la consola el menú contextual del modo debug	
9	En la consola, ingresar a la opción ver archivo de registro	Visualizar entradas con valores de temperatura con saltos regulares en la marca de tiempo	



<b>PROYECTO:</b>		<b>EAMMRA</b>	
ID CASO DE PRUEBA:	UC02	Test diseñado por:	Patricio Bos
NOMBRE DEL CASO:	Cambio de período de adquisición de temp.	Fecha de diseño:	01/10/18
Versión de firmware:	1.0	Test ejecutado por:	Patricio Bos
		Fecha de ejecución:	01/11/18
Pre condiciones:	Al menos 1 sensor DS18B20 conectado al puerto 1-WIRE (GPIO3[0]) de la CIAA-NXP.		
Post condiciones:	ÉXITO: Cambio en el intervalo entre marcadas de tiempo de las mediciones. FALLA: Sin cambio en el intervalo entre marcadas de tiempo de las mediciones.		
Resumen del Test:	Test de cambio de configuración en el período de adquisición de valores de temperatura. Después de energizar el sistema, se inicializan los módulos y se comienza a registrar en forma periódica valores de temperatura. Al ingresar un nuevo valor para el período de adquisición se debe observar un cambio proporcional entre las marcas de tiempo de los datos registrados en la tarjeta microSD.		
Step #	Descripción	Resultado Esperado	Resultado Obtenido
1	Energizar el sistema	Mensaje de startup en la consola	
2	(sistema) Inicialización de todos los módulos	Mensajes de inicialización de cada módulo en la consola	
3	Esperar a que se cumpla 1 minuto desde el paso #1	-	
4	En la consola, ingresar al modo configuración. Luego ingresar al módulo de adquisición 1-WIRE	Visualizar en la consola el menú contextual del configuración del módulo de adquisición	
5	En la consola, ingresar a la opción cambiar configuración del archivo de registro. Luego ingresar 2000 ms	Visualizar entradas con valores de temperatura con saltos regulares en la marca de tiempo	
6	(sistema) Se encola un mensaje para el módulo de adquisición con la señal sig_config y el nuevo valor para el período de adquisición.	-	
7	(sistema) El módulo de adquisición recibe la señal sig_config y un nuevo valor de período. Cuando expira el nuevo timeout, el módulo de adquisición envía una señal sig_write con un valor de temperatura al módulo de almacenamiento.	-	
8	En la consola, ingresar al modo debug	Visualizar en la consola el menú contextual del modo debug	
9	En la consola, ingresar a la opción ver archivo de registro	Visualizar entradas con valores de temperatura con saltos regulares en la marca de tiempo	

