

ESPECIFICACIÓN TÉCNICA

Comunicación Modbus TCP

Relés de Protección ABB Serie 615
Subestación CELTA 1

Documento de referencia para desarrollo de sistema de monitoreo

Versión 1.0 - 1/1/2026

1. Información General de la Red

1.1 Parámetros de conexión

Parámetro	Valor
Protocolo	Modbus TCP
Puerto	502 (todos los equipos)
Red	172.16.0.x
Función de lectura	03 - Read Holding Registers
Offset de registros	Manual ABB - 1 = Código Modbus (ej: manual 173 = código 172)

1.2 Inventario de equipos

Ubicación	Modelo	IP	CT	Mide tensión
Alimentador 1	REF615	172.16.0.1	1A	No
Alimentador 2	REF615	172.16.0.2	1A	No
Alimentador 3	REF615	172.16.0.3	1A	No
Alimentador 4	REF615	172.16.0.4	1A	No
Alimentador 8	REF615	172.16.0.8	1A	No
TERNA 3	REF615	172.16.0.11	5A	Sí (HBFF)
TERNA 4	REF615	172.16.0.12	5A	Sí (HBFF)
TRAFO 1 - Salida 33kV	REF615	172.16.0.6	5A	No
TRAFO 1 - Dif 13.2kV	RET615	172.16.0.5	5A	Sí
TRAFO 2 - Salida 33kV	REF615	172.16.0.9	5A	No
TRAFO 2 - Dif 13.2kV	RET615	172.16.0.7	5A	Sí
TRAFO 3 - Dual	RET615	172.16.0.10	5A	Sí (HBTF)

2. Registros Comunes a Todos los Equipos

Los siguientes registros están disponibles en todos los relés REF615 y RET615 y deben ser la base del sistema de monitoreo.

2.1 Registros de Estado del Sistema (SSR)

Código	Manual	Nombre	Descripción
127	128	SSR1	Salud del dispositivo (Ready)
128	129	SSR2	Modo del IED (Local/Remoto, Setting Group)
129	130	SSR3	Data pendiente (eventos, fallas sin leer)
131	132	SSR5	Alive counter (heartbeat)

2.1.1 Registro 127 - SSR1 (Salud del dispositivo)

Este registro indica si el LED Ready está encendido (equipo funcionando correctamente).

Bit	Nombre	Valores	Acción si activo
0	Error global	0=OK, 1=Error	ALERTA CRÍTICA
1	Warning global	0=OK, 1=Warning	ALERTA WARNING

Lógica de interpretación:

```
def interpretar_ssrl(valor):
    error = (valor >> 0) & 1
    warning = (valor >> 1) & 1

    if error:
        return {'estado': 'ERROR', 'ready': False, 'alerta': 'CRITICA'}
    elif warning:
        return {'estado': 'WARNING', 'ready': True, 'alerta': 'WARNING'}
    else:
        return {'estado': 'OK', 'ready': True, 'alerta': None}

# Ejemplo: valor = 0 → Ready encendido, todo OK
```

2.1.2 Registro 128 - SSR2 (Modo del IED)

Bit	Nombre	Valores	Interpretación
0	Test mode	0=Normal, 1=Test	Modo prueba activo
2	Local/Remoto	0=Remoto, 1=Local	Control local activo
3-5	Setting Group	Valor binario + 1	Grupo de ajustes activo (1-6)
8	Cold start	1=Último reset fue arranque frío	Informativo

Lógica de interpretación:

```
def interpretar_ssrr2(valor):
    return {
        'test_mode': (valor >> 0) & 1,
        'modo_local': (valor >> 2) & 1,
        'setting_group': ((valor >> 3) & 0x07) + 1,
        'cold_start': (valor >> 8) & 1
    }

# Ejemplo: valor = 270 (binario 100001110)
# test_mode=0, modo_local=1, setting_group=1, cold_start=1
```

2.1.3 Registro 129 - SSR3 (Data pendiente)

Indica si hay eventos o fallas sin leer. Útil para saber si el equipo requiere atención.

Bit	Nombre	Valores	Acción si activo
0	Eventos sin leer	0=No, 1=Sí	Informativo
1	Fallas sin leer	0=No, 1=Sí	REVISAR
4	Bit momentáneo actualizado	0=No, 1=Sí	Cambio reciente
5	MCD activo (sin reconocer)	0=No, 1=Sí	LED titilando
6	Device restart	0=No, 1=Sí	El equipo se reinició

2.2 Registro 172 - Estado de LEDs Programables

Este es el registro más importante para detectar alarmas visualmente. Cada bit representa un LED del panel frontal.

NOTA: El valor 0 significa que no hay ningún LED de alarma encendido. Cualquier valor distinto de 0 indica que hay al menos un LED activo.

Decodificación de bits:

Bit	LED	Cálculo
0	LED 1	(valor >> 0) & 1 → Si es 1, LED 1 encendido
1	LED 2	(valor >> 1) & 1 → Si es 1, LED 2 encendido
2	LED 3	(valor >> 2) & 1 → Si es 1, LED 3 encendido
...
10	LED 11	(valor >> 10) & 1 → Si es 1, LED 11 encendido

Función completa de decodificación:

```
def decodificar_leds(valor, etiquetas_equipo):
    """
    valor: entero leído del registro 172
    etiquetas_equipo: diccionario {bit: 'descripción'}
    """

    leds_activos = []
    for bit in range(16): # 16 bits máximo
        if (valor >> bit) & 1:
            etiqueta = etiquetas_equipo.get(bit, f'LED {bit+1}')
            leds_activos.append({
                'led': bit + 1,
                'bit': bit,
                'etiqueta': etiqueta
            })
    return leds_activos

# Ejemplo: valor = 17 (binario 10001)
# LEDs activos: bit 0 (LED 1) y bit 4 (LED 5)
```

Valores de ejemplo observados:

Valor	Binario	Significado
0	0000 0000 0000	Sin alarmas - todo OK
17	0000 0001 0001	LED 1 y LED 5 encendidos (típico: Arranque I> y Arranque lo>)
1024	0100 0000 0000	Solo LED 11 (típico: Posición CB cerrado)
1041	0100 0001 0001	LED 1, 5 y 11 (Alarms + posición CB)

2.3 Registro 170 - Start / Trip

Indica si hay un arranque de protección (Start) o si hubo un disparo (Trip).

Bit	Nombre	Valores	Acción si activo
0	Start general	0=No, 1=Arranque	Protección detectó algo
1	Start MCD (sin reconocer)	0=Reconocido, 1=Sin reconocer	LED Start titilando
2	Trip (Operate)	0=No, 1=Disparo	¡HUBO DISPARO!
3	Trip MCD (sin reconocer)	0=Reconocido, 1=Sin reconocer	LED Trip titilando

Lógica de interpretación:

```
def interpretar_start_trip(valor):
    return {
        'start': (valor >> 0) & 1,
        'start_sin_reconocer': (valor >> 1) & 1,
        'trip': (valor >> 2) & 1,
        'trip_sin_reconocer': (valor >> 3) & 1
    }

def generar_alerta_start_trip(datos):
    if datos['trip']:
        return 'ALERTA_CRITICA: ¡DISPARO ACTIVO!'
    elif datos['trip_sin_reconocer']:
        return 'ALERTA_ALTA: Disparo sin reconocer'
    elif datos['start']:
        return 'ALERTA_MEDIA: Arranque de protección'
    elif datos['start_sin_reconocer']:
        return 'INFO: Arranque sin reconocer'
    return None
```

2.4 Registros 137-139 - Corrientes de Fase

Valores de corriente instantánea de las tres fases. Disponible en todos los equipos.

Código	Manual	Descripción	Fórmula
137	138	IL1 (Fase A)	valor / 1000 = corriente en xIn
138	139	IL2 (Fase B)	valor / 1000 = corriente en xIn
139	140	IL3 (Fase C)	valor / 1000 = corriente en xIn

💡 NOTA: El valor viene en milésimas de In. Por ejemplo, si leés 285, la corriente es $0.285 \times In$ (28.5% de la nominal).

Función de conversión:

```
def leer_corrientes(client, slave_id):
    regs = client.read_holding_registers(address=137, count=3, slave=slave_id)
    return {
        'IL1': regs.registers[0] / 1000, # En xIn
        'IL2': regs.registers[1] / 1000,
        'IL3': regs.registers[2] / 1000
    }

# Ejemplo: [285, 282, 280] → IL1=0.285, IL2=0.282, IL3=0.280 xIn
```

3. Registros Específicos por Tipo de Equipo

3.1 Equipos con Medición de Tensión (HBFF, RET615)

Solo disponible en: TERNA 3, TERNA 4, TRAFO 1 Dif, TRAFO 2 Dif, TRAFO 3

Código	Manual	Descripción	Fórmula
151	152	Tensión fase A	valor / 1000 = tensión en xUn
152	153	Tensión fase B	valor / 1000 = tensión en xUn
153	154	Tensión fase C	valor / 1000 = tensión en xUn
154	155	Tensión línea A-B	valor / 1000 = tensión en xUn
155	156	Tensión línea B-C	valor / 1000 = tensión en xUn
156	157	Tensión línea C-A	valor / 1000 = tensión en xUn

3.2 Potencias (Solo TERNA 3 y TERNA 4)

Código	Manual	Descripción	Unidad/Fórmula
160-161	161-162	Potencia activa P	32 bits con signo [W o kW según config]
162-163	163-164	Potencia reactiva Q	32 bits con signo [VAr o kVAr]
164-165	165-166	Potencia aparente S	32 bits con signo [VA o kVA]
166	167	Factor de potencia	valor / 1000 (ej: 946 = 0.946)

3.3 Corriente Residual Io (RET615)

Solo disponible en: TRAFO 1 Dif, TRAFO 2 Dif, TRAFO 3

Código	Manual	Descripción	Fórmula
141	142	Corriente residual Io	valor / 1000 = corriente en xIn

3.4 Registro 175 - Posición del Interruptor

Este registro requiere que el interruptor tenga conectada la retroalimentación de posición al relé.

Valor	Bits activos	Significado
768	Bits 8 y 9	Sin retroalimentación de posición (celda vieja)
769	Bits 0, 8 y 9	Interruptor CERRADO (bit 0 = Close)
770	Bits 1, 8 y 9	Interruptor ABIERTO (bit 1 = Open)
256	Bit 8	Configuración diferente (TERNAAs)

Función de interpretación:

```
def interpretar_posicion_cb(valor):
    close = (valor >> 0) & 1
    open_bit = (valor >> 1) & 1 # 'open' es palabra reservada

    if close and not open_bit:
        return 'CERRADO'
    elif open_bit and not close:
        return 'ABIERTO'
    elif close and open_bit:
        return 'ERROR: Posición inválida (ambos bits)'
    else:
        return 'INDETERMINADO: Sin retroalimentación'
```

4. Etiquetas de LEDs por Equipo

Mapeo de bits del registro 172 a las etiquetas físicas de cada equipo. Usar estas tablas para traducir los bits activos a descripciones legibles.

4.1 Alimentadores (1, 2, 3, 4, 8)

Bit	Etiqueta del LED
0	Arranque I>
1	Disparo I>
2	Falla a Tierra sensible / Disparo I>>
3	Disparo I>>
4	Arranque lo >
5	Disparo Falla a Tierra
6	Desbalance de Fases
7	Recierre Habilitado
8	Recierre en Progreso
9	Pos CB Abierto
10	Pos CB Cerrado

4.2 TERNA 3 y TERNA 4

Bit	Etiqueta del LED
0	Sobreintensidad
1	Falta a tierra
2	Sobre/sub tensión
3	Desbalance de fases
4	Sobrecarga térmica
5	Fallo de interruptor
6	Disparo reg. perturb.
7	Monitorización interruptor
8	Supervisión

4.3 TRAFO 1 y 2 - Diferencial (RET615)

Bit	Etiqueta del LED
0	Prot dif pol. etapa baja
1	Prot. dif. etapa alta
2	Sobreintensidad
3	Falta a tierra restringida
4	Falta a tierra
5	Fallo de interruptor
6	F. sec. neg. / sobrecarga 1°
7	Disparo reg. perturb.
8	Supervisión
9	Disparo externo

5. Lógica de Alertas para el Sistema de Monitoreo

5.1 Niveles de alerta

Nivel	Color sugerido	Condiciones
CRÍTICA	Rojo	Trip activo, Error global, Fallo de interruptor
ALTA	Naranja	Trip sin reconocer, Disparo I>/I>>/Io, Prot diferencial
MEDIA	Amarillo	Start activo, Arranque I>/Io, Warning global, Reinicio
INFO	Azul	Eventos sin leer, MCD activo, Modo local
OK	Verde	Ready encendido, sin LEDs de alarma, comunicación OK

5.2 Algoritmo de evaluación

```
def evaluar_estado_rele(datos):
    """
    datos: diccionario con valores de registros
    Retorna: nivel de alerta y lista de mensajes
    """

    alertas = []
    nivel_maximo = 'OK'

    # 1. Verificar salud (registro 127)
    if datos.get('ssrl_error'):
        alertas.append('Error global del dispositivo')
        nivel_maximo = 'CRITICA'
    elif datos.get('ssrl_warning'):
        alertas.append('Warning global del dispositivo')
        nivel_maximo = max_nivel(nivel_maximo, 'MEDIA')

    # 2. Verificar Trip (registro 170)
    if datos.get('trip'):
        alertas.append(';DISPARO ACTIVO!')
        nivel_maximo = 'CRITICA'
    elif datos.get('trip_sin_reconocer'):
        alertas.append('Disparo sin reconocer')
        nivel_maximo = max_nivel(nivel_maximo, 'ALTA')

    # 3. Verificar Start (registro 170)
    if datos.get('start'):
        alertas.append('Arranque de protección activo')
        nivel_maximo = max_nivel(nivel_maximo, 'MEDIA')

    # 4. Verificar LEDs (registro 172)
    if datos.get('leds_valor', 0) != 0:
        leds = decodificar_leds(datos['leds_valor'], datos['etiquetas'])
        for led in leds:
            # Clasificar según etiqueta
            if 'Disparo' in led['etiqueta'] or 'Fallo' in led['etiqueta']:
                alertas.append(f"LED {led['led']}: {led['etiqueta']}")
                nivel_maximo = max_nivel(nivel_maximo, 'ALTA')
            elif 'Arranque' in led['etiqueta']:
                alertas.append(f"LED {led['led']}: {led['etiqueta']}")
                nivel_maximo = max_nivel(nivel_maximo, 'MEDIA')
            elif 'Pos CB' not in led['etiqueta']: # Ignorar posición CB
                alertas.append(f"LED {led['led']}: {led['etiqueta']}")
                nivel_maximo = max_nivel(nivel_maximo, 'INFO')
```

```
return {'nivel': nivel_maximo, 'alertas': alertas}
```

6. Resumen de Registros para Monitoreo

6.1 Lectura mínima recomendada (todos los equipos)

Registro	Nombre	Propósito	Frecuencia
127	SSR1	Estado Ready	Cada 5-10 segundos
129	SSR3	Eventos/fallas pendientes	Cada 5-10 segundos
131	SSR5	Alive (heartbeat)	Cada 5-10 segundos
170	Start/Trip	Detección de disparos	Cada 1-5 segundos
172	LEDs	Alarmas visibles	Cada 1-5 segundos
137-139	Corrientes	Monitoreo de carga	Cada 5-30 segundos

6.2 Lectura extendida (equipos con tensión)

Registro	Nombre	Propósito	Equipos
151-156	Tensiones	Monitoreo de tensión	TERNAs, TRAFOs Dif
160-166	Potencias	P, Q, S, FP	TERNAs
141	Io	Corriente residual	RET615 (TRAFOs Dif)

6.3 Estructura de datos sugerida

```
EQUIPOS = {
    'ALIM_1': {
        'ip': '172.16.0.1',
        'modelo': 'REF615',
        'tiene_tension': False,
        'tiene_io': False,
        'etiquetas_led': {
            0: 'Arranque I>',
            1: 'Disparo I>',
            4: 'Arranque Io >',
            5: 'Disparo Falla a Tierra',
            10: 'Pos CB Cerrado'
            # ... completar según documentación
        }
    },
    'TERNA_3': {
        'ip': '172.16.0.11',
        'modelo': 'REF615',
        'tiene_tension': True,
        'tiene_io': False,
        'etiquetas_led': {
            0: 'Sobreintensidad',
            1: 'Falta a tierra',
            2: 'Sobre/sub tensión',
            # ... completar según documentación
        }
    },
    # ... resto de equipos
}
```

7. Notas Importantes

7.1 Offset de registros

⚠️ ALERTA: Siempre restar 1 al número de registro del manual ABB para obtener la dirección Modbus correcta. Ejemplo: Manual dice 173 → usar 172 en código.

7.2 Valores de corriente y tensión

Los valores vienen multiplicados por 1000. Dividir por 1000 para obtener el valor en xIn o xUn (relativo a la nominal configurada en el relé).

7.3 Heartbeat / Comunicación

El registro 131 (SSR5 - Alive counter) cambia constantemente. Si deja de cambiar durante varias lecturas consecutivas, hay un problema de comunicación con el equipo.

7.4 Diferencia entre celdas

Las celdas "viejas" (Alimentadores 1, 3, 4, TRAFOS Salida) no tienen retroalimentación de posición del interruptor. El registro 175 mostrará 768 en lugar de 769/770. El registro 174 mostrará 64 en lugar de 16.

Característica	Celda Vieja	Celda Nueva
Registro 174	64	16
Registro 175	768	769 (cerrado) / 770 (abierto)
LED Pos CB	Siempre apagado	Refleja estado real