

# RESUMEN COMPLETO - PROTOCOLO MODBUS MPPT XY7025

## Información General

- **Protocolo:** MODBUS-RTU
- **Velocidad por defecto:** 115200 bps
- **Dirección por defecto:** 1
- **Funciones soportadas:** 0x03, 0x06, 0x10
- **Formato:** 8 bits datos, 1 bit parada, sin paridad

## MAPEO DE REGISTROS (Decimal y Hexadecimal)

### Registros de Configuración y Control

Dirección Dec	Dirección Hex	Nombre	R/W	Unidad	Decimales	Descripción
0	0x0000	V-SET	R/W	V	2	Voltaje configurado
1	0x0001	I-SET	R/W	A	3	Corriente configurada
15	0x000F	LOCK	R/W	-	0	Bloqueo teclado (0=OFF, 1=ON)
16	0x0010	PROTECT	R	-	0	Estado protecciones
17	0x0011	CVCC	R	-	0	Modo carga (0=CV, 1=CC)
18	0x0012	ONOFF	R/W	-	0	Estado salida (0=OFF, 1=ON)
19	0x0013	F-C	R/W	-	0	Escala temperatura
20	0x0014	B-LED	R/W	-	0	Brillo pantalla (0-5)
21	0x0015	SLEEP	R/W	M	0	Tiempo apagado pantalla
22	0x0016	MODEL	R	-	0	Modelo equipo
23	0x0016	VERSION	R	-	0	Version equipo
24	0x0018	SLAVE-ADD	R/W	-	0	Dirección dispositivo
25	0x0019	BAUDRATE_L	R/W	-	0	Velocidad comunicación
26	0x001A	T-IN-OFFSET	R/W	F/C	1	Corrección de temperatura interna
27	0x001B	T-EX-OFFSET	R/W	F/C	1	Corrección de temperatura externa
28	0x001C	BUZZER	R/W	-	0	Interruptor de timbre
29	0x001D	EXTRACT-M	R/W	-	0	Perfil predefinido (0-9)
30	0x001E	DEVICE	R/W	-	0	Estado del equipo
31	0x001F	MPPT-SW	R/W	-	0	Habilitar MPPT
32	0x0020	MPPT-K	R/W	-	0	Factor calibración
33	0x0021	BatFul	R/W	A	1	Corriente fin de carga

## Registros de Monitoreo (Solo Lectura)

Dirección Dec	Dirección Hex	Nombre	Unidad	Decimales	Descripción
2	0x0002	VOUT	V	2	Voltaje salida
3	0x0003	IOUT	A	3	Corriente salida
4	0x0004	POWER	W	2	Potencia salida
5	0x0005	UIN	V	2	Voltaje entrada
6-7	0x0006-7	AH-LOW/HIGH	mAh	0	Acumulado AH
8-9	0x0008-9	WH-LOW/HIGH	mWh	0	Acumulado WH
10-12	0x000A-C	OUT_H/M/S	H/M/S	0	Tiempo activo
13-14	0x000D-E	T_IN/T_EX	°C/°F	1	Temperaturas

## Registros de Monitoreo 10 Perfiles (R/W)

Dirección Dec	Dirección Hex	Nombre	Unidad	Decimales	Descripción
88-89	0x0058-9	S-OAH_LOW/HIGH	mAh	0	Contador de Amperios-hora
90-91	0x005A-B	S-OWH_LOW/HIGH	10 mWh	0	Contador de Wattios-hora

## Registros de Protección 10 Perfiles (R/W)

Dirección Dec	Dirección Hex	Nombre	Unidad	Decimales	Descripción
80	0x0050	V-SET	V	2	Voltaje de salida Perfil rápido
81	0x0051	I-SET	A	2	Corriente de salida Perfil rápido
82	0x0052	S-LVP	V	2	Umbral subtensión
83	0x0053	S-OVP	V	2	Umbral sobretensión
84	0x0054	S-OCP	A	3	Umbral sobrecorriente
85	0x0055	S-OPP	W	1	Umbral sobrepotencia
86-87	0x0056-7	S-OHP_H/M	H/M	0	Límite tiempo operación
92	0x005C	S-OTP	°C/°F	0	Temp. máxima interna
93	0x005D	S-INI	-	0	Estado inicial salida
94	0x005E	S-ETP	-	0	Sensor temp. externo

# FUNCIONES MODBUS SOPORTADAS

## Función 0x03 - Leer Registros

```
// Leer 2 registros empezando en 0x0002 (VOUT e IOUT)
uint8_t resultado = mppt.readHoldingRegisters(0x0002, 2);
if(resultado == mppt.ku8MBSuccess) {
    float voltaje = mppt.getResponseBuffer(0) / 100.0;
    float corriente = mppt.getResponseBuffer(1) / 1000.0;
}
```

## Función 0x06 - Escribir Registro Simple

```
// Configurar voltaje a 24.00V (0x0960 = 2400)
uint8_t resultado = mppt.writeSingleRegister(0x0000, 2400);
```

## Función 0x10 - Escribir Múltiples Registros

```
// Configurar V-SET e I-SET atómicamente
mppt.setTransmitBuffer(0, 2400); // 24.00V
mppt.setTransmitBuffer(1, 15000); // 15.000A
uint8_t resultado = mppt.writeMultipleRegisters(0x0000, 2);
```

# PERFILES PREDEFINIDOS (EXTRACT-M)

## Grupos de Perfiles M0-M9

```
// Cambiar a perfil M2 (LiFeP04)
mppt.writeSingleRegister(0x001D, 2);

// Direcciones base de cada perfil:
// M0: 0x0050, M1: 0x0060, M2: 0x0070, ..., M9: 0x00E0
uint16_t dirBasePerfil = 0x0050 + (numeroPerfil * 0x0010);
```

# CÓDIGOS DE ERROR COMUNES

0x00 Éxito	Operación completada
0xE1 Timeout	Verificar conexión
0xE2 CRC inválido	Revisar cableado/ruido
0x01 Función no soportada	Usar solo 0x03, 0x06, 0x10

0x02 Dirección inválida	Verificar mapa de registros
0x03 Valor inválido	Verificar rango de datos

## CONFIGURACIÓN RECOMENDADA ARDUINO

```
#include <ModbusMaster.h>
#include <SoftwareSerial.h>

ModbusMaster mppt;
#define DE_RE_PIN 4

void setup() {
  Serial2.begin(115200);
  mppt.begin(1, Serial2);

  // Configurar control RS485
  pinMode(DE_RE_PIN, OUTPUT);
  mppt.preTransmission([](){
    digitalWrite(DE_RE_PIN, HIGH);
  });
  mppt.postTransmission([](){
    digitalWrite(DE_RE_PIN, LOW);
  });
}
```

## EJEMPLOS PRÁCTICOS

### Leer energía acumulada

```
uint32_t leerEnergiaTotal() {
  if(mppt.readHoldingRegisters(0x0008, 2) == mppt.ku8MBSuccess) {
    uint16_t low = mppt.getResponseBuffer(0);
    uint16_t high = mppt.getResponseBuffer(1);
    return (high << 16) | low; // Devolver en mWh
  }
  return 0;
}
```

### Configuración segura

```
bool configuracionSegura(float voltaje, float corriente) {
  if(voltaje < 10.0 || voltaje > 30.0) return false;
  if(corriente < 0.1 || corriente > 30.0) return false;

  mppt.setTransmitBuffer(0, voltaje * 100);
  mppt.setTransmitBuffer(1, corriente * 1000);
  return (mppt.writeMultipleRegisters(0x0000, 2) == mppt.ku8MBSuccess);
}
```

# CONCLUSIÓN FINAL

Con este resumen puedes:

1. **Monitorear** todos los parámetros del MPPT
2. **Configurar** voltaje, corriente y protecciones
3. **Controlar** el funcionamiento del equipo
4. **Integrar** con Arduino/ESP32
5. Guardar todos los registros de monitoreo y poder graficarlos