M4x4_R3 librería runtime

Documentación del firmware y uso

Attila Zoltan Tamas
athomas
tomasati67@gmail.com
2025.

Indice

1. Introducción

1.1 Objetivo

La librería **M4x4_R3** proporciona la capa de control necesaria para el hardware **MOLTINO 4x4**, siempre que la placa de control utilizada sea un **Arduino UNO R3** con procesador **ATmega328**.

Esta librería permite operar el sistema de manera autónoma bajo el protocolo **MODBUS RTU** mediante comunicación **RS485**. Ofrece soporte para:

- 4 entradas digitales con funciones avanzadas,
- 4 entradas analógicas de 16 bits,
- 4 salidas digitales mediante relés,
- 4 salidas analógicas con rango de **0 a 10 V**.

El funcionamiento básico puede implementarse a través de un sketch de Arduino sencillo, invocando únicamente dos funciones principales. La librería está distribuida en forma **precompilada,** pero todas las funciones de gestión de entradas/salidas y las adicionales se exponen como **funciones públicas**, lo que permite al usuario desarrollar programas tipo PLC que supervisen y controlen las señales de entrada y salida de forma flexible y dependiente entre sí.

El uso de las funciones públicas garantiza el acceso a todas las capacidades avanzadas que la librería ofrece, manteniendo la compatibilidad y ampliando las posibilidades de desarrollo sobre la plataforma. La programación se realiza con el entorno Arduino IDE 2.

1.2 Distribución y soporte

La librería **M4x4_R3** se entrega exclusivamente en **formato binario precompilado**. La ampliación funcional debe realizarse mediante el uso de las **funciones públicas** desde sketches externos. Las actualizaciones y notas de versión se publicarán junto con los binarios correspondientes.

1.3 Requisito de entorno de desarrollo

Para la correcta utilización del archivo precompilado de la librería *M4x4_R3*, es imprescindible emplear el **Arduino IDE en su versión 2.x o superior**. Otras versiones del entorno de desarrollo no garantizan compatibilidad ni un funcionamiento adecuado del firmware.

2. Instalación de la librería M4x4_R3

La librería M4x4_R3 se distribuye únicamente en formato precompilado y requiere la instalación previa de tres dependencias externas.

2.1 Descarga de la librería M4x4_R3

1. Visita la página oficial de GitHub de la librería M4x4_R3: github.com/Moltino
2. Haz clic en el botón **“Code”** y selecciona **“Download ZIP”**.
3. Guarda el archivo en tu equipo.

2.2 Instalación de dependencias

Antes de instalar la librería **M4x4_R3**, asegúrate de tener instaladas las siguientes librerías mediante el **Library Manager** del Arduino IDE:

- **Adafruit_ADS1X15** → necesaria para la gestión del conversor ADC
- **digitalWriteFast** → necesaria para manejar las entradas y salidas.
- **WDT** → necesaria para el sketch Arduino

Procedimiento de instalación:

1. Abre el **Arduino IDE**.
2. Ve a **Programa → Incluir librería → Gestionar bibliotecas...**.
3. Busca *Adafruit ADS1X15* y haz clic en **Instalar.**
4. Busca *digitalWriteFast* y haz clic en **Instalar.**
5. Busca *WDT* y haz clic en **Instalar**

2.3 Instalación de M4x4_R3

1. Abre el **Arduino IDE**.
2. Ve a **Programa → Incluir librería → Añadir librería .ZIP...**.
3. Selecciona el archivo `.zip` descargado de GitHub y haz clic en **Abrir**.
4. El IDE añadirá la librería **M4x4_R3** al entorno de desarrollo.

2.4 Verificación de instalación

- Abre el menú **Programa → Incluir librería**.

- Verifica que **M4x4_R3**, **Adafruit_ADS1X15, digitalWriteFast y WDT** aparecen en la lista. Si todas están presentes, la instalación ha sido exitosa.

2.5 Uso en el proyecto

Para usar la librería en tu sketch principal, únicamente es necesario incluir:

#include <M4x4_R3.h>

Las dependencias (`Adafruit_ADS1X15 y digitalWriteFast`) son gestionadas internamente por **M4x4_R3**, por lo que no es necesario incluirlas manualmente.

3. Funciones principales

El firmware está basado en el estándar **MODBUS RTU**, garantizando compatibilidad en la comunicación y en la interpretación de comandos. No obstante, incorpora una funcionalidad extendida: los comandos también pueden ejecutarse de forma local, sin necesidad de recibirlos a través de la línea de comunicación.

Gracias a esta característica, es posible desarrollar **sketches independientes** en Arduino que permiten al dispositivo operar de manera autónoma. La programación se centra principalmente en asignar valores a registros y leer los resultados mediante las funciones expuestas en la **API pública** de la librería. Todo ello está soportado por el firmware **M4x4_R3**.

Cuando se utiliza en modo **MODBUS RTU**, el sistema se comporta como un equipo plenamente integrado, aprovechando todas las capacidades que ofrece el hardware **MOLTINO 4x4** junto con el firmware **M4x4_R3**.

Nota sobre direcciones Modbus en Moltino:
Todas las direcciones siguen el estándar Modbus: offsets base 0.
Por ejemplo:

- Holding register número 0 → se documenta como **40000** en algunos SCADA.
- Holding register número 1 → se documenta como **40001** en algunos SCADA.

3.1 Identificación del dispositivo, comunicación

El dispositivo **MOLTINO 4x4** utiliza el estándar **MODBUS RTU** sobre red **RS-485** para el intercambio de datos.

Cada unidad dispone de un **número de identificación único (ID de esclavo)** que permite enviar y recibir mensajes a través de la red MODBUS.

- El rango permitido para la identificación es de **1 a 246**.
- El valor por defecto es **1**.

- En caso de programar un valor fuera de rango, el firmware lo sustituirá automáticamente por **1**.

Dirección universal de programación

El identificador **247** es reconocido por todos los dispositivos MOLTINO 4x4. Se emplea únicamente como recurso para configurar el equipo cuando su identificación individual se desconoce.

> **Nota:** aunque la dirección 247 permite programar todos los dispositivos simultáneamente, se recomienda **evitar su uso**, ya que puede producir conflictos y comportamientos no deseados. Esto se debe a que todos los dispositivos responderán de forma simultánea al maestro, generando posibles errores en la comunicación.

Comunicación en difusión (Broadcast)

El MOLTINO 4x4 admite comandos en **modo broadcast**, utilizando como dirección el identificador **0**. En este caso, todos los dispositivos ejecutan el comando recibido, pero **no generan respuesta** hacia el maestro.

Los comandos compatibles en modo broadcast son:

- `05dec – WRITE SINGLE COIL`
- `06dec – WRITE SINGLE REGISTER`
- `15dec – WRITE MULTIPLE COILS`
- `65dec – RESET COUNTERS`

Registro de configuración de identificación

El número de identificación del dispositivo se configura en el **registro HOLDING 40033**.

3.2 Puertos de comunicación

El **MOLTINO 4x4** incorpora de forma nativa soporte para comunicación **RS-485**, cuya circuitería está integrada en la placa base. La librería **M4x4_R3** implementa el protocolo estándar **MODBUS RTU** sobre este puerto.

Adicionalmente, la librería **M4x4_R3** permite el intercambio de datos a través del **puerto USB** de la placa **Arduino UNO R3** instalada en la placa base del MOLTINO.

> **Importante:** el puerto USB **no suministra alimentación** al dispositivo. Es obligatorio proporcionar alimentación externa mediante:

- **13,4 VDC** a través del conector de la placa Arduino, o
- **24 VDC** mediante las bornas de alimentación del MOLTINO.

La selección del modo de comunicación (RS-485 o USB) se realiza mediante el **jumper de configuración** ubicado en la placa base.

- El puerto USB se utiliza únicamente para intercambio de datos y para la programación de la placa Arduino UNO R3.

Parámetros de comunicación serie

La configuración por defecto es:

- **Formato:** 8-N-1 (8 bits de datos, sin paridad, 1 bit de parada).
- **Velocidad:** 9600 bps.

Velocidades soportadas

Las velocidades de transmisión y recepción admitidas por la librería **M4x4_R3** se configuran a través del **registro HOLDING 40034**, asignando el código decimal correspondiente:

Velocidad (bps)	Código en registro 40034
9600 *(por defecto)*	96
19200	19
38400	38
57600	57
74800	74
115200	115
230400	230

3.3 Comandos MODBUS, funciones API

El **MOLTINO 4x4** con la librería **M4x4_R3** acepta los siguientes comandos MODBUS:

- Read Coils (0x01)
- Read Discrete Inputs (0x02)
- Read Holding Registers (0x03)
- Read Input Register (0x04)
- Write Single Coil (0x05)
- Write Single Register (0x06)
- Write Multiple Coils (0x0F)
- Reset Counters (0x41)

Cada comando representa una función que se puede llamar en un sketch Arduino entregarselo el control al dispositivo en cuanto la tareja se permite.

Funciones API:

void begin(); *Sirve para inicializar y configurar el Arduino adecuadamente para el hardware. MOLTINO 4x4*

void dataMonitoring(); C*ontrola la comunicación estandar* ***MODBUS***

void readCounters(); *Controla el tiempo runtime, registra los impulsos llegados en las entradas digitales.*

void sendTextLcd(); *Opcional, es para controlar la pantalla cuando este incorporada.*

uint16_t readCoils(const uint16_t start, uint16_t quantity); *Devuelve el estado de los reles. (ej: coil1, coil4 están activadas: B0000 0000 0000 1001)*

uint16_t readDiscreteInputs(const uint16_t start, uint16_t quantity); *Devuelve los estados instantaneos y latcheadas de las entradas digitales según el registro llamado.*

uint16_t readInputReg(const uint16_t address); *Leer el valor en las entradas analógicas 16bit, una a la vez.*

uint16_t readHoldingReg(const uint16_t address); *Sirve para leer los valores de las salidas analogicas, los contadores 32bits, la frequencia de cada entrada digital, el error de los bucles de corriente 4-20mA y, las configuraciones de MOLTINO.*

void writeSingleCoil(const uint16_t address, uint16_t value); Activar *o deactivar los reles uno por uno.*

void writeSingleReg(const uint16_t address, uint16_t value); *Escribir los registros HOLDING uno por uno. (P.e.: configurar el dispositivo, cambiar la tension de las salidas analogicas)*

void writeMultipleCoils(uint16_t outputs_value); *Activar o apagar los reles en grupo.*

void resetCounters(const uint16_t register32, uint16_t Count); *Poner a 0 a los contadores. El value „register32" es la dirección del primer registro, „Count" es la cantidad del „register32" que son deseados a borrar.*

3.4 Configuraciones de MOLTINO 4x4 á través de la librería runtime M4x4_R3

Las configuraciones de MOLTINO igual son eficientes cuando el dispositivo es un miembro de la red MODBUS o funciona de forma independiente automaticamente. La region de la configuración se empieza con la dirección Holding registro 40033 y llega a su final en la dirección 40047. Todas las direcciones de los registros MOLTINO se encuentran registradas en la mapa de registros, igual como en el arhivo M4x4_R3.h están definidos con nombres para facilitar el uso de las direcciones en el sketch de los usuarios. Los valores de las configuraciones están guardadas en la memória EEPROM.

Dirección	Función
40033	ID de esclavo
40034	velocidad serial
40035	mascara de las entradas digitales (PNP/NPN)
40036	memória de los contadores (activar - deactivar)
40037	mascara de dirección de conteo (ascendente/descendente)
40038	escala de valor de salida de las entradas analogicas
40039	reservado
40040	4-20mA error bit/canal (encender-apagar)
40041	reservado
40042	escala de valor de salida de las salidas analogicas
40043	Forma de medición de las entradas analogicas proyectada en la pantalla lcd (points,V,mA)
40044	demora de rele 1
40045	demora de rele 2
40046	demora de rele 3
40047	demora de rele 4

3.5 Detalles de configuración

3.5.1 Mascara de las entradas digitales

La máscara permite configurar individualmente el modo de activación de cada entrada digital. Se utilizan los 4 bits menos significativos del byte menos significados.

- Si un bit está en **1**, la entrada correspondiente se activa con **flanco ascendente (PNP)**.
- Si un bit está en **0**, la entrada correspondiente se activa con **flanco descendente (NPN)**.

Tabla de verdad

ENTRADA	MASK	SALIDA
1	1	1
0	1	0
1	0	0
0	0	1

3.5.2 Memoria de contadores

Cada entrada digital del **MOLTINO 4x4** está asociada a un contador de impulsos de **32 bits**, capaz de procesar señales de hasta **15 kHz**. El firmware incluye una función de respaldo que permite conservar el valor de los contadores, evitando la pérdida de datos en caso de apagado del dispositivo.

La configuración de esta función es binaria:

- **0** → Memoria desactivada (los valores de los contadores no se guardan).
- **1** → Memoria activada (los valores de los contadores se almacenan y se restauran automáticamente tras un reinicio).

Cuando la memoria está habilitada, el sistema recupera los valores previos de los contadores después de un reinicio, garantizando la continuidad de las tareas como si no hubiera ocurrido ninguna interrupción en el funcionamiento.

3.5.3 Dirección de conteo (ascendente/descendente)

La mascara se permite configurar a cada contador individualmente. Se utilizan los 4 bits menos significativos del byte menos significados.

- Bit 0 : conteo ascendente
- bit 1 : conteo descendente

Correspondencia:

- bit0 → contador1
- bit1 → contador2
- bit2 → contador3
- bit3 → contador4

3.5.4 Escala de valor de salida de entradas analógicas

El sistema permite recalibrar la escala de salida de cada entrada analógica, principalmente por razones de compatibilidad. La resolución del conversor A/D es de 16 bits, donde el bit más significativo indica la polaridad. En consecuencia, el valor máximo representable por canal es 32767.

La configuración se realiza mediante un **registro de 16 bits**, que define el valor máximo de la escala de salida. Este registro puede tomar valores de **0 a 65535**:

- Valor por defecto: **32.767**.
- Valor configurado: se convierte en el nuevo máximo de la escala de salida.

El método de recalibración ajusta la salida a una escala comprendida entre **0** y el valor definido en el registro, admitiendo también valores inferiores al predeterminado. Cabe destacar que esta configuración **no modifica la resolución física del conversor**, únicamente redefine el rango de salida lógico.

3.5.5 4-20mA error bit/canal (encender-apagar)

Se utilizan los 4 bits menos significativos del byte menos significados. Su función es activar o deactivar los bits que indican si la corriente en el canal que corresponde es menos que 4mA.

Estados:

- 1: el bit indicador activado
- 0: el bit indicador desactivado

Los canales correspondientes empiezan desde el bit menos significado. El bit0 corresponde al bit indicador de canal 1 y, así sucecivamente.

3.5.6 Escala de valor de salida de las salidas analogicas

El metodo es igual que en las entradas analogicas, salvo que en este caso lo utiliza para reescalar el control de las salidas analogicas. La resolución fisica es 8bit, pero por la compatibilidad es capaz de resolver valores de control hasta 16bit.

3.5.7 Forma de medición de las entradas analogicas proyectada en la pantalla lcd

La pantalla lcd es opcional, pero cuando este incorporado facilita comprobar a los **valores instantaneos** y las configuraciones actuales. Á través de esta configuración se ajusta la unidad de medida en el que muestra la pantalla el valor de la medición de las **entradas analogicas**.

Hay 3 formas de medición:

- **Tensión (Volt) hasta 3 decimales**
- **Corriente (mA) hasta 3 decimales**
- **Puntos que salen directamente del conversor AD**

Configuración:

Canales	*4*		*3*		*2*		*1*		
Bit / byte	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**	
puntos	0	0	0	0	0	0	0	0	**0: todos los canales proyectados en puntos**
Valor (dec)	0		0		0		0		
puntos	0	1	0	1	0	1	0	1	**85:todos los canales proyectados en puntos**
Valor (dec)	64		16		4		1		
Volt	1	0	1	0	1	0	1	0	**250:todos los canales proyectados en Voltios**
Valor* (dec)	128		32		8		2		
mA	1	1	1	1	1	1	1	1	**255:todos los canales proyectados en mA**
Valor (dec)	192		48		12		3		

La configuración se permite que cada canal aparezca en diferente unidades de medición. Para calcular el valor de la configuración hay que sumar los valores decimales que aparecen en la tabla. Por ejemplo:

Canal 1: Voltios → 2

Canal 2: mA → 12

Canal 3: puntos → 0

Canal 4: Voltios → 128

+_____

Config. valor: 142dec = 1000 1110

3.5.8 Demora de relés

El **MOLTINO 4x4** dispone de **4 salidas digitales** implementadas mediante relés, los cuales pueden configurarse en dos modos de operación:

- **Biestable**
- **Monoestable**

La configuración de los relés permite:

- Definir el **modo de funcionamiento**.
- Ajustar un **tiempo de demora** comprendido entre **1 segundo** y **18 horas, 12 minutos y 15 segundos**.

El comportamiento es el siguiente:

- Si el tiempo de demora se establece en **0**, el relé correspondiente opera en **modo biestable**.
- Si el tiempo configurado es **mayor que 0**, el relé se activa y se desactiva automáticamente una vez transcurrido el tiempo de demora.

Cada relé dispone de un **registro de 16 bits**, donde **cada bit equivale a 1 segundo**, lo que permite definir de forma precisa la duración de la temporización.

4. Uso en Sketch Arduino

El **MOLTINO 4x4** permite a cualquier usuario desarrollar firmware personalizado adaptado a necesidades específicas. La librería **runtime M4x4_R3** se utiliza invocándola desde un **sketch de Arduino**.

En la carpeta **examples** se incluyen varios ejemplos prácticos que ilustran el uso correcto de la librería.
Para obtener un funcionamiento básico, basta con **copiar y ejecutar el sketch de ejemplo `Moltino4x4_R3`**.

Dentro del sketch, el usuario debe llamar a las **funciones API** de la librería **M4x4_R3**, y procesar los datos obtenidos utilizando las funciones estándar de Arduino.

4.1 Archivo `M4x4_R3.h`

El archivo **`M4x4_R3.h`**, ubicado en la carpeta **`M4x4_R3/src`**, contiene todos los elementos necesarios para programar un sketch con la librería runtime.

Este archivo incluye:

- Definiciones de direcciones internas mediante **nombres simbólicos** para facilitar su identificación y lectura en el código.
- **Prototipos** de todas las funciones API.
- Definiciones de los textos mostrados en la pantalla LCD

4.2 Personalización de la pantalla LCD

El dispositivo incorpora una pantalla LCD de **2 líneas x 16 caracteres**. La información se organiza de la siguiente manera:

- **Línea superior**: indica la entrada/salida a la que corresponde el valor mostrado.
- **Línea inferior**: muestra la dirección numerada y el valor con su unidad de medida.

Los valores son proporcionados por la librería precompilada. Sin embargo, la **línea superior es personalizable**, permitiendo modificar el texto directamente en el archivo **`M4x4_R3.h`**.

Limitaciones y recomendaciones:

- La pantalla solo admite **16 caracteres por línea**; cualquier carácter adicional no será visible.
- Se recomienda **no modificar la configuración predeterminada**, salvo en el caso de personalizar los nombres mostrados.