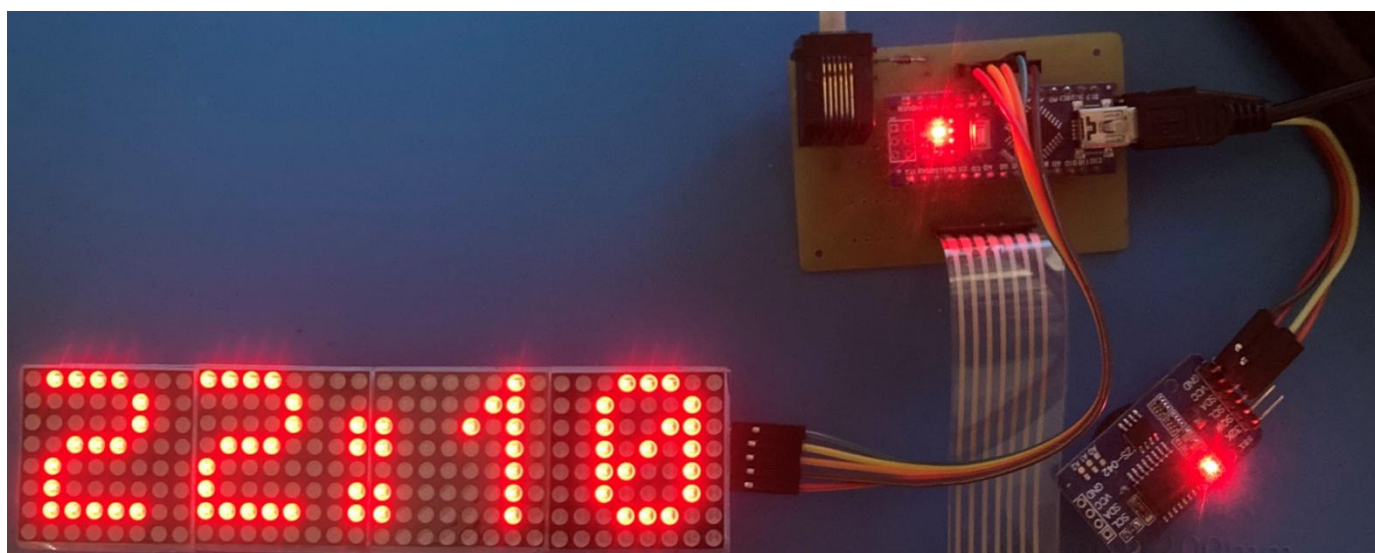


Manual de usuario

FastClockLN by Paco



v0.4 01/2024

Índice

Introducción.....	3
1. Instrucciones de seguridad.....	3
2. Garantía	3
3. Soporte y ayuda	3
4. Antes del montaje	4
5. Conexión	4
6. Menú	5
6.1 Modo	5
6.2 Hora	6
6.3 Reloj	6
6.4 Ratio.....	7
6.5 Brillo.....	7
Anexo I: Esquema	8
Anexo II: Lista de componentes	11
Anexo III: Programación con Arduino IDE	12
Anexo IV: Consejos	16

El software usa librerías con diferentes licencias, siga todos sus términos incluidos en ellas. Loconet es una marca registrada de Digitrax. Arduino es una marca registrada de Arduino S.r.l.

Introducción

FastClockLN surgió de la necesidad de tener un sencillo display para el reloj rápido (*Fast Clock*) de Loconet para un grupo modular como proyecto *Do-It-Yourself* (DIY).

NO está permitida su comercialización.

Basado en un Arduino Nano (también Uno o Pro Mini), dispone de cuatro displays de matriz de puntos 8x8 con el MX7219, cuatro botones y un módulo de reloj en tiempo real (RTC) con el DS3231.

1. Instrucciones de seguridad

Este producto no es conveniente para niños menores de 14 años. Como puede ser ingerido por un niño menor de 3 años se debe mantener fuera de su alcance.

Un uso inadecuado puede implicar riesgo de lesiones debido a bordes afilados y puntas que pinchen.

Se recomienda la utilización de un soldador del tipo lápiz con una punta muy fina para evitar daños a los componentes.

2. Garantía

Toda la información para la construcción y uso del **FastClockLN** al ser un diseño "Hazlo tu mismo" (*DIY - Do it Yourself*) se proporciona "tal cual", sin garantía de ningún tipo, expresa o implícita, incluidos aquellos daños producidos por mala interpretación, montaje, uso o manipulación inadecuada.

3. Soporte y Ayuda

La información para la construcción y/o uso del mando **FastClockLN** se halla disponible en los sitios siguientes dónde es posible, sin estar asegurada ni tener compromiso alguno, obtener ayuda:

<https://usuaris.tinet.cat/fmco>

4. Antes del montaje

En los Anexos de este documento encontrará los esquemas del **FastClockLN** para montarlo.

Para su programación se usa el entorno Arduino IDE desde el cual es posible cargar el programa y las librerías necesarias.

En los Anexos de este documento se describe como programar el Arduino para nuestro **FastClockLN**.

5. Conexión

Al conectar el **FastClockLN** a la central mediante el interface Loconet o dar tensión (según se monte) aparecerá la pantalla de bienvenida indicando el tipo de central al que está conectado y el modo de operación mediante textos deslizantes. Finalmente se mostrará la hora del sistema.



Los puntos de la hora ':' parpadearán si el tiempo del sistema está corriendo (ratio de 1:1 a 127:1) y permanecerán fijos si esta pausado (ratio 0:1).

Si la central está sin tensión en las vías (*Stop*), **FastClockLN** mostrará en pantalla el correspondiente mensaje. Esto no afecta al cómputo de la hora interna, cuando vuelva la tensión a las vías (*Go*) se mostrará la hora del sistema que corresponda.



6. Menú

FastClockLN dispone de cuatro teclas para su configuración:

'MENU'	Muestra el menú. Al volver a pulsar muestra la hora del sistema
'HORA / +'	Incrementa las horas o el valor. Selecciona a la siguiente opción.
'MIN / -'	Incrementa los minutos. Decrementa el valor. Selecciona la anterior opción.
'ENTER'	Selecciona o guarda el valor mostrado de la opción actual.

Al pulsar la tecla 'MENU' se mostrará el menú de opciones de **FastClockLN**, con las teclas 'HORA / +' y 'MIN / -' podemos mostrar las diferentes opciones disponibles. Con la tecla 'ENTER' seleccionamos la opción mostrada para modificarla.

Mientras estemos en el menú, el tiempo del sistema sigue corriendo (si el ratio no es 0).

6.1 Modo

FastClockLN puede funcionar en diferentes modos (Esclavo, Sync, RTC Sync y Master), seleccione con las teclas 'HORA / +' y 'MIN / -' el adecuado para usted. Pulsando 'ENTER' se guardará el valor y permanecerá para posteriores arranques.

MODO ESCLAVO

Este es el modo básico, simplemente mostrará la hora que transmite el sistema. Si el ratio transmitido no es 0 (no está en pausa) actualizará su reloj interno de acuerdo al ratio y lo irá mostrando en pantalla. Se sincronizará con el sistema cuando este transmita la hora del sistema por el bus.

Este modo no hace uso del módulo de reloj en tiempo real (RTC).

MODO SYNC

Este modo es como el modo Esclavo pero aproximadamente cada 60 segundos preguntará (SYNC) al sistema por el tiempo actual para sincronizar todos los relojes.

Sólo un dispositivo del bus puede generar el SYNC, el resto permanecerán en modo esclavo. Este modo no hace uso del módulo de reloj en tiempo real (RTC).

MODO RTC SYNC

Este modo es como el modo Sync pero al recibir tensión enviará la hora del módulo de reloj en tiempo real (RTC) con un ratio de 1:1 al sistema, luego pasará a modo Sync automáticamente.

Recuerde que sólo un dispositivo del bus puede generar el SYNC, los demás son esclavos. Este modo se usa para funcionar con ratio 1:1 con hora real cuando la central soporta el Slot del Fast Clock (DR5000, Intellibox II, ..).

MODO MASTER

Este modo esta previsto principalmente para cuando la central no soporta el *Slot* del *Fast Clock* (TwinCenter,..) o no tenemos central y montamos una red para los relojes (se necesitaría un circuito de pull-up de 15mA, ver Anexo IV) aunque también se puede usar para funcionar con hora real (RTC) 1:1 con mayor precisión que algunas centrales.

Al seleccionar este modo se establece el ratio en 1:1 si el RTC está montado y se pausa (ratio 0:1) si no lo está.

Al dar tensión, si el RTC DS3231 está montado el reloj arranca automáticamente con la hora real y cada minuto enviará al bus la hora del RTC. Si se selecciona el ratio 1:1 teniendo RTC también se actualizará con la hora real del RTC. Si el RTC no está montado, al dar tensión el reloj permanecerá pausado mostrando 0:00, desde el menú podremos ajustar la hora y el ratio. Cada nuevo minuto del reloj a escala se enviará por el bus la hora interna.

En este modo no es necesario que otro dispositivo envíe el SYNC. Tenga en cuenta que sólo un dispositivo del bus puede ser Master (normalmente es la central).

Si **FastClockLN** está funcionando en modo Master se puede pausar el tiempo fácilmente si se pulsa la tecla 'MIN / -' mientras se muestra la hora del sistema. Para reanudar el cómputo del tiempo desde la hora pausada pulse la tecla 'HORA / +'. Si se tiene montado el RTC tiene la opción de reanudar el tiempo desde la hora real actual entrando a la opción Ratio del menú y cambiando el ratio de Pausa a 1:1.

6.2 Hora

La hora del sistema se puede cambiar con esta opción. Use la tecla 'HORA / +' para cambiar las horas y la tecla 'MIN / -' para cambiar los minutos. Al pulsar la tecla 'ENTER' se enviará la nueva hora al sistema.

6.3 Reloj

La hora del reloj en tiempo real (RTC) se puede cambiar con esta opción. Use la tecla 'HORA / +' para cambiar las horas y la tecla 'MIN / -' para cambiar los minutos. Al pulsar la tecla 'ENTER' se guardará en el RTC la nueva hora (los segundos se establecerán en 0).

Si el RTC no está montado se mostrará el siguiente mensaje:



6.4 Ratio

FastClockLN permite establecer un ratio para el reloj a escala desde 1:1 a 127:1 o pausar el reloj (ratio 0:1). No hay un ratio más correcto que otro, varía con la escala de la maqueta y de usuario a usuario pero lo habitual es usarlo entre 1:1 (tiempo real) a 12:1 (una hora de juego representan 12 horas en la maqueta).

Pulsando 'HORA / +' se aumenta el valor, con la tecla 'MIN / -' se decrementa el valor. Al pulsar 'ENTER' se establecerá el nuevo ratio del sistema.

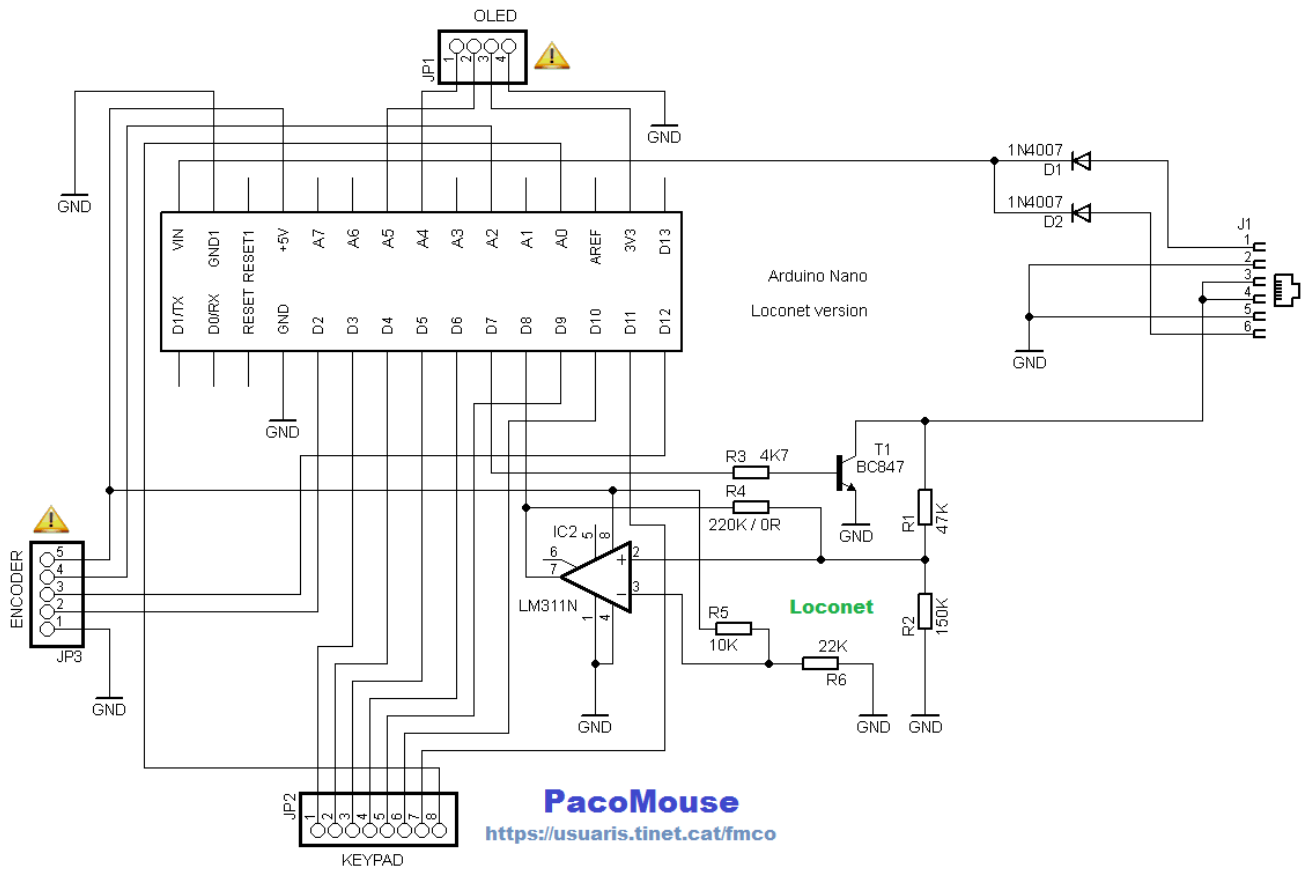
6.5 Brillo

Se puede cambiar el brillo de la pantalla del **FastClockLN** entre un valor de 0 y 15 para adaptar la visibilidad a las circunstancias ambientales.

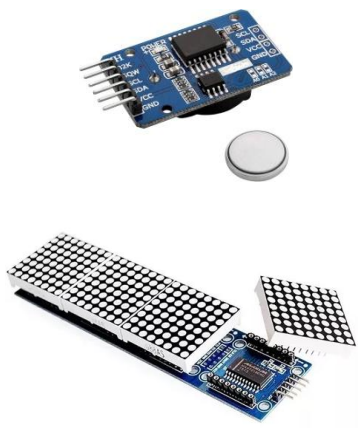
Pulsando 'HORA / +' se aumenta el valor y el brillo de la pantalla, pulsando 'MIN / -' se decrementa el valor y el brillo de la pantalla. Cuando tengamos el brillo adecuado a nuestro entorno, pulsando 'ENTER' se guardará el valor y permanecerá para posteriores arranques.

Anexo I: Esquema

Esquema Arduino Nano (Usando la placa de PacoMouse)

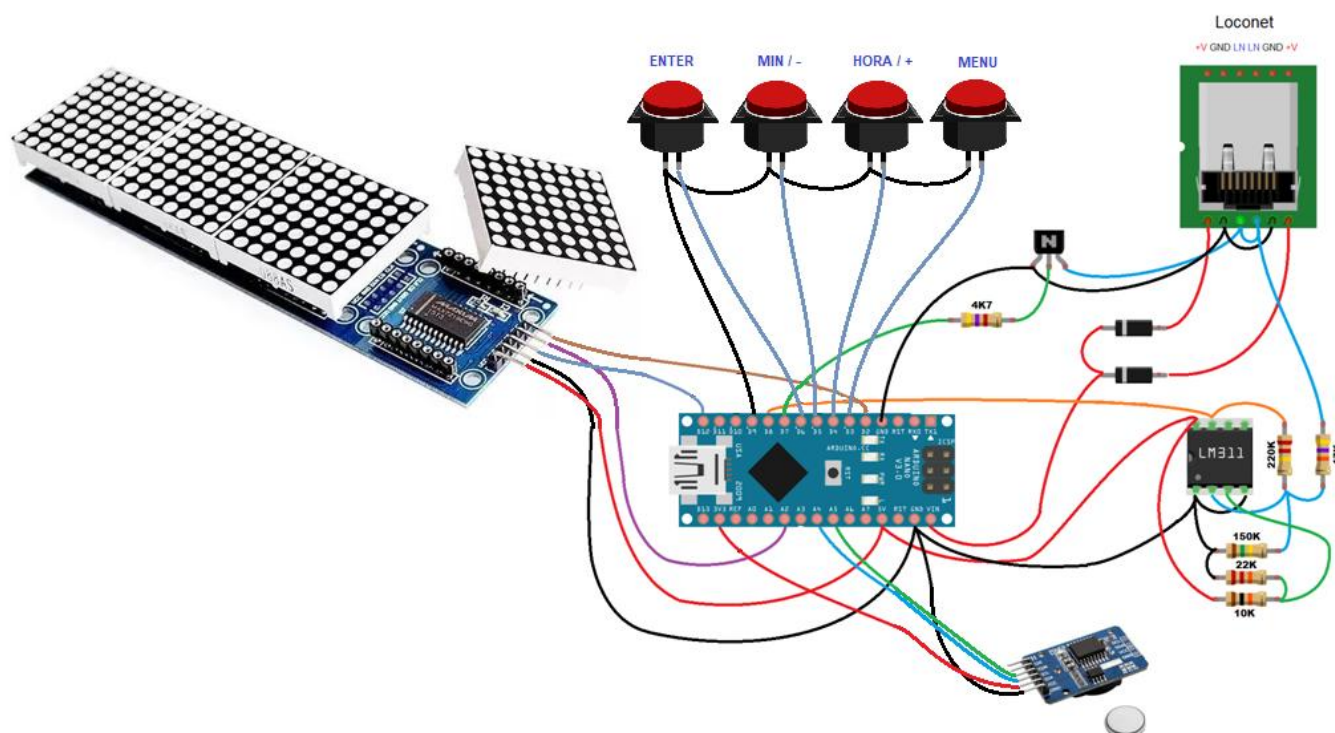


 **Comprobar orden conexión pins**

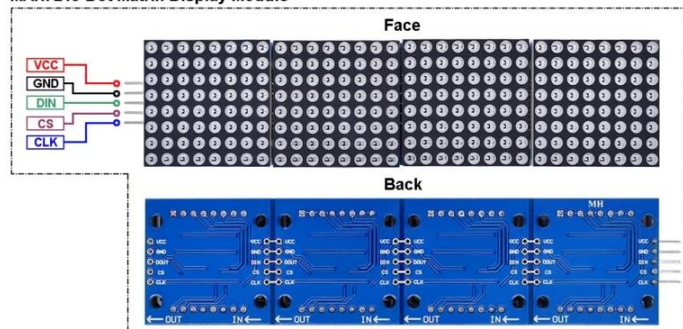


DS3231	Arduino (JP1)
GND	GND
VCC	3V3
SCL	A5
SDA	A4

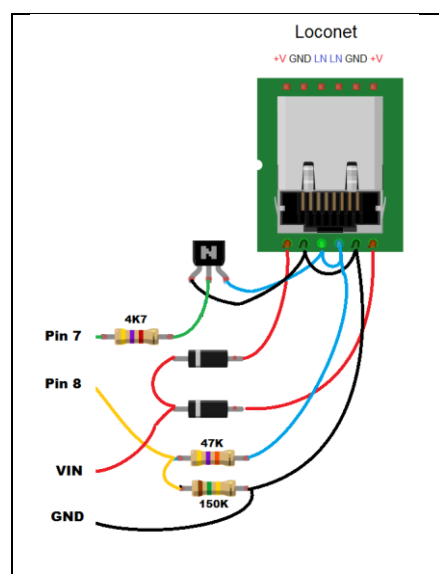
MX7219	Arduino (JP3)
GND	GND
CLK	D2
DIN	D12
CS	A2
VCC	+5V



MAX7219 Dot Matrix Display Module



Display tipo FC16



Versión interface Loconet simple

NOTA: Comprobad las conexiones de vuestro hardware.

Conexión de componentes

Estas son las conexiones del Arduino Nano con los diferentes elementos.

Arduino Nano	MX7219 *		Teclas	DS3231	Observaciones
	SPI	PacoMouse			
TX					
RX					
D2		CLK			
D3			FILA1		
D4			FILA2		
D5			FILA3		
D6			FILA4		
D7					Loconet: Resistencia del BC547
D8					Loconet: LM311 pin 7
D9			COL1		Común de las teclas
D10	CS				
D11	DIN				
D12		DIN			
D13	CLK				
A0					
A1					
A2		CS			
A3					
A4				SDA	
A5				SCL	
5V	VCC	VCC		VCC	
3V3					
GND	GND	GND		GND	

(*) Según la conexión utilizada, uso del interface SPI o placa **PacoMouse**, modificar la configuración de las opciones de usuario en el programa (ver Anexo III)

Anexo II: Lista de componentes

	<i>Observaciones</i>
Arduino Nano/Pro mini (ATmega328)	
Teclas (x4)	
DS3231	Módulo reloj I2C
Transistor: BC547	
Resistencias: 150K/47K/4K7 22K/10K/220K *	
Diodo: 1N4007 **	Si se alimenta desde el bus
LM311 *	
RJ12	
Display 8x8 MX7219 (x4) **	

- (*) Componentes adicionales para un interface clásico Loconet.
- (**) El consumo del display MX7219 es elevado si se usa un brillo alto. En este caso es preferible alimentar desde el USB del Arduino Nano y no montar los diodos.

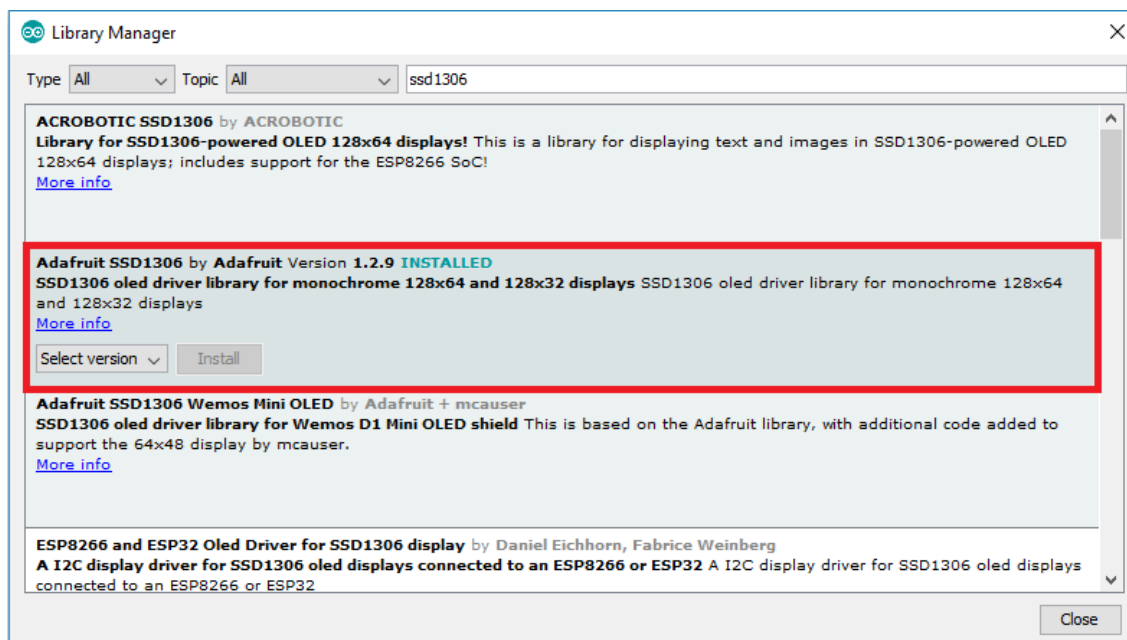
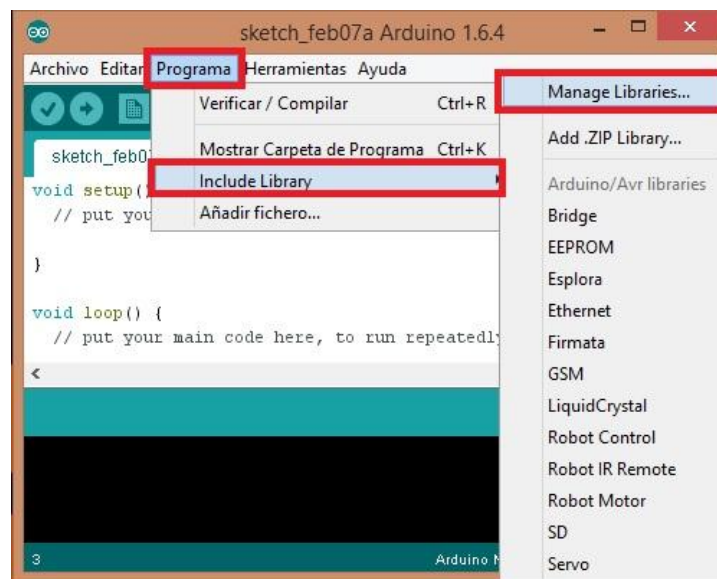
Anexo III: Programación con Arduino IDE

Programa

Se puede usar tanto en un Arduino Nano como un Arduino Nano o un Arduino Pro Mini.

Desde el mismo Arduino IDE (<https://www.arduino.cc/en/Main/Software>) se carga el programa en la placa Arduino, a través de la conexión USB del mismo.

Previamente tenemos que instalar las librerías **MD_MAX72xx.h** (v3.5.1), **Loconet.h** (1.1.9), **uRTCLib.h** (v6.6.1) y **Keypad.h** (v3.1.1) con la opción *Programa -> Incluir Librería -> Gestionar Librerías*



Configuración

Hay una serie de configuraciones que se pueden escoger cuando se programa el **FastClockLN** desde Arduino IDE en función de lo que tengamos montado.

ATENCIÓN: Modifique únicamente la sección de código que empieza con:

```
////////////////////////////////////  
// ***** USER OPTIONS *****  
////////////////////////////////////
```

y finaliza con:

```
////////////////////////////////////  
// ***** END OF USER OPTIONS *****  
////////////////////////////////////
```

A continuación se indican para cada opción los valores que se pueden **modificar** en color **rojo** y los posibles valores en color **violeta**:

NOTA: Para **descomentar** una línea, eliminar los caracteres **//** al inicio de la misma.

Cuando haya adaptado las opciones a su **FastClockLN** ya podrá pasar al capítulo siguiente para programar su Arduino.

Este software y los archivos asociados son un proyecto de bricolaje que no está diseñado para uso comercial.

EL SOFTWARE SE PROPORCIONA "TAL CUAL", SIN GARANTÍA DE NINGÚN TIPO, EXPRESA O IMPLÍCITA.

Las fuentes solo se proporcionan para compilar y cargar en el dispositivo.
No se permite modificar el código fuente ni derivar/publicar este proyecto.

EL USO COMERCIAL ESTÁ PROHIBIDO.

This software and associated files are a DIY project that is not intended for commercial use.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED.

Sources are only provided for building and uploading to the device.
You are not allowed to modify the source code or fork/publish this project.

COMMERCIAL USE IS FORBIDDEN.

Opciones de display MX7219

TIPO DE DISPLAY

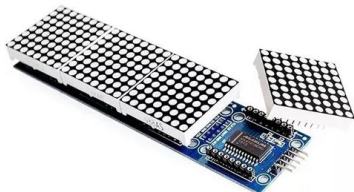
FastClockLN muestra la información en un display de cuatro módulos matriz de puntos 8x8 controlados por el MX7219. Hay diferentes tipos de estos módulos, cambie esta opción para una correcta visualización en sus módulos.

```
/*          ///< DR - Digits as rows; CR - Columns Reversed; RR - Rows Reversed
  GENERIC_HW, ///< Use 'generic' style hardware modules commonly available.
  FC16_HW,    ///< Use FC-16 style hardware module.
  PAROLA_HW,  ///< Use the Parola style hardware modules.
  ICSTATION_HW, ///< Use ICStation style hardware module.

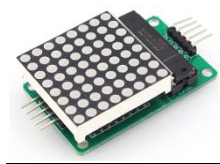
  DR0CR0RR0_HW, ///< Structured name
  DR0CR0RR1_HW, ///< Structured name
  DR0CR1RR0_HW, ///< Structured name; equivalent to GENERIC_HW
  DR0CR1RR1_HW, ///< Structured name
  DR1CR0RR0_HW, ///< Structured name; equivalent to FC16_HW
  DR1CR0RR1_HW, ///< Structured name
  DR1CR1RR0_HW, ///< Structured name; equivalent to PAROLA_HW
  DR1CR1RR1_HW, ///< Structured name; equivalent to ICSTATION_HW
*/

#define HARDWARE_TYPE MD_MAX72XX::FC16_HW // Tipo modulo
```

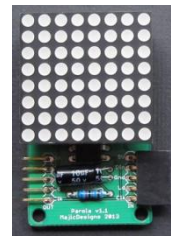
NOTA: Puede usar el programa de ejemplo **MD_MAX72xx_HW_Mapper.ino** de la librería **MD7219xx.h** para averiguar su tipo de módulo. Defina los pines que usa y siga las instrucciones en el Monitor Serie.



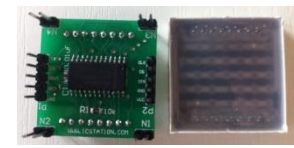
FC16



GENERIC



PAROLA



IC STATION

Opciones de conexión

Placa PacoMouse

Si hemos montado el **FastClockLN** en una placa **PacoMouse** con Arduino Nano tenemos que comentar la línea correspondiente al interface SPI que debe quedar así:

```
// #define USE_SPI // Descomentar si se usa pins SPI standard (mas rapido) en lugar de la placa PacoMouse
```

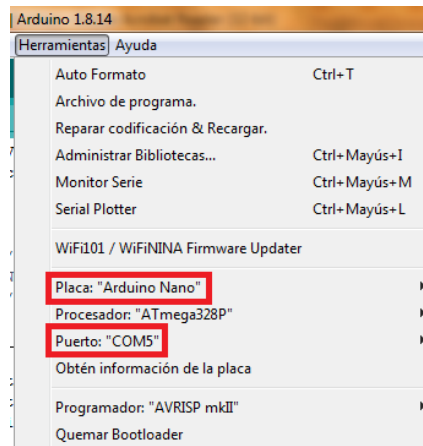
Interface SPI

Si hemos montado el **FastClockLN** con Arduino Nano (Uno o Pro Mini) y queremos usar el bus SPI tenemos que descomentar la línea correspondiente al interface SPI que debe quedar así:

```
#define USE_SPI // Descomentar si se usa pins SPI standard (mas rapido) en lugar de la placa PacoMouse
```

Programación del Arduino

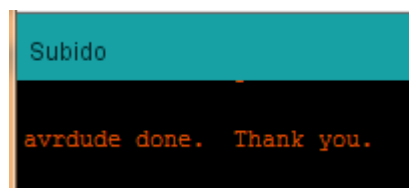
Una vez hemos abierto el sketch y actualizado las opciones según nuestras preferencias y hardware, en el Arduino IDE comprobamos que hemos seleccionado la placa Arduino que tenemos y el puerto al que está conectada. Quizás necesitemos instalar algún driver previamente para que nos aparezca listado el puerto al que está conectado.



Para cargar el programa pulsaremos sobre el botón de 'Subir' programa.



Tras compilar el programa, si todo va bien nos informará de que ha sido subido.



Anexo IV: Consejos

AL CONECTAR POR PRIMERA VEZ

FastClockLN guarda las diferentes opciones escogidas por el usuario en la memoria EEPROM del controlador. La primera vez que se conecte **FastClockLN** la memoria EEPROM contendrá valores indeterminados por lo que es posible que algunas opciones no sean las preferidas por el usuario. Para solucionarlo pulse la tecla 'MENU' y seleccione las opciones de **Modo** y **Brillo** que mejor se adapten a sus necesidades.

CONSUMOS DEL DISPLAY

Los displays de matriz de puntos usan LED distribuidos en 8 líneas, de 8 puntos por cada módulo. Para **FastClockLN** necesitamos cuatro de estos módulos por lo que el consumo es alto, mayor cuanto más brillo:

Visualización en FastClockLN	Consumo medio (Brillo: 0)	Consumo medio (Brillo: 15)
Hora	30mA	185mA
Mensaje deslizante	30mA	200mA
Todos los puntos encendidos	90mA	300mA

ATENCIÓN: Si se va a usar con brillos superiores a 1 es mejor alimentar el **FastClockLN** desde el conector USB del Arduino. En este caso no instalar los dos diodos 1N4007 para alimentarse desde el bus, o bien usar un cable RJ12 de 4 hilos en lugar del habitual RJ12 de 6 hilos del bus.

FAST CLOCK

Si la central soporta el *Fast Clock* en el slot 123 (Intellibox II, DR5000, etc.) elegiremos el modo **Esclavo**, **Sync** o **RTC Sync** que mejor se adapte a nuestras necesidades. Si no soporta el *Fast Clock* (como la TwinCenter, Daisy II, etc.) o no usamos una central elegiremos el modo **Master**.

Si no se usan los modos **RTC Sync** o **Master** con ratio 1:1 de precisión en el **FastClockLN** no se necesita montar el módulo DS3231, **FastClockLN** detectará esta configuración y se adaptará a ella.

En caso de no usar central, debemos proporcionar un *pull-up* de 15mA al bus para que funcione la red de **FastClockLN**:

