

Industrial communication solutions for Windows

XLI4278 Driver Manual

Motorola LI-4278 BarCode Reader Serial Driver

Contents

XLI4278 technical specifications	2
General information.....	2
Command list	4
EsperarCodigo	4
Error messages.....	4
Supported devices.....	5

CPKSoft Engineering

Industrial communication drivers.

www.cpksoft.com

www.facebook.com/cpksoftengineering

cpksoftengineering@

hotmail.com

phone: 54-911-45788354

1990-2015

Industrial communication solutions for Windows

XLI4278 technical specifications

General information

El driver XLI4278 permite comunicarse con los lectores de código de barra Motorola LI-4278 mediante una comunicación de tipo serie como alternativa a la comúnmente utilizada emulación de teclado que viene establecida de fábrica.

CONEXION:

Si la base se conecta mediante USB, previamente se la debe configurar para que trabaje en modo 'Simple COM Port Emulation' y además se debe tener instalado en la PC el emulador de puerto serie virtual denominado 'Symbol COM Port Emulation Driver v 1.8.5.zip' y que esta disponible en el siguiente link:

<https://atgsupportcentral.motorolasolutions.com/ewa/pub/getFile.do?fileName=/ssi/emb/downloads/Symbol%20COM%20Port%20Emulation%20Driver%20v%201.8.5.zip>

Los pasos para configurar la base en modo 'Simple COM Port Emulation' se encuentran en la pagina 178 del manual del equipo, que se puede descargar del siguiente link:

http://www.posmicro.com/downloads/li4278_manual.pdf

FUNCIONAMIENTO:

Este driver comienza poniéndose en modo 'escucha', quedando atento a lo que el lector le envía a la PC y sin transmitir nada hacia el lector. El principio de funcionamiento entonces es que al activar el driver (mediante HMITalk o mediante NetTalk), el mismo se queda directamente esperando a que le llegue un código de barras, sin enviar previamente ningún mensaje.

Para detectar la llegada de un código se utilizan dos mecanismos: por tiempo entre caracteres y/o por longitud del código. En casos como el EAN13 se puede utilizar la detección por longitud ya que los códigos son siempre fijos de 13 caracteres. En casos como el EAN128 los códigos pueden ser de longitud variable por lo que se debe aplicar la detección por timeout entre caracteres. No obstante, ambos mecanismos de detección pueden utilizarse combinadamente en casos especiales (terminar por longitud o por tiempo, lo que ocurra antes).

El timeout entre caracteres, definido en DriverP1, se utiliza para establecer una condición de terminación de los telegramas basada en tiempo, para todos aquellos casos en los que no es aplicable establecer una longitud de telegrama específica a ser esperada. El tiempo que se utilice en el timeout debe ser del orden de algunos milisegundos, mayor que el tiempo que pueda haber entre dos porciones de un mismo código (por ejemplo si estuviera siendo tipeado remotamente carácter por carácter por un operador o si el lector lo subdividiera en partes al transmitirlo). A su vez debe ser menor que el tiempo que pueda haber entre dos códigos consecutivos, para que las lecturas puedan ser correctamente individualizadas como dos códigos diferentes. Un valor adecuado podría rondar los 100 a 500 ms.

En el caso que se utilice detección por timeout entre caracteres, lo que se evalúa es el tiempo transcurrido entre la recepción de un carácter individual y la recepción del siguiente carácter individual, de manera tal que cuando se haya recibido un carácter y haya pasado el timeout entre caracteres establecido sin recibirse otro carácter, se considerará que se ha recibido un mensaje o

CPKSoft Engineering

Industrial communication drivers.

www.cpksoft.com

www.facebook.com/cpksoftengineering

cpksoftengineering@hotmail.com

cpksoftengineering@hotmail.com

phone: 54-911-45788354

1990-2015

Industrial communication solutions for Windows

codigo completo desde el lector, dando por finalizada la lectura y retornando los datos del codigo a la aplicacion. Este mecanismo permite detectar multiples tipos de codigo al mismo tiempo.

Los mecanismos de deteccion que utiliza el driver no dependen del formato propio de cada tipo de codigo por lo que es aplicable a todos los casos donde el lector transmita espontaneamente una secuencia de caracteres alfanumericos por cada codigo, separados un tiempo uno del otro (UPC/EAN: UPC-A, UPC-E, UPC-E1, EAN-8/JAN 8, EAN-13/JAN 13, Bookland EAN, Bookland ISBN, Código Extendido de cupones UCC, Codigo ISSN EAN 128 incluyendo GS1- 128, ISBT 128, Concatenacion ISBT, Codigo 39 incluyendo Codigo 39 Trioptico, Conversion Codigo 39 a Codigo 32 (Codigo farmaceutico de Italia), Codigo 39 Full ASCII Conversion, codigo 93 Codigo 11, Matriz 2 de 5, Intercalado 2 de 5 (ITF), Diferenciado 2 de 5 (DTF), Codabar (NW – 7), MSI Chino 2 de 5, IATA Inverso 1D (salvo todos los DataBars GS1) GS1 DataBar, incluyendo GS1 DataBar-14, GS1 DataBar Limitado, GS1 DataBar Extendido). Queda luego a cargo de la aplicacion la interpretacion del contenido de cada codigo recibido.

Por otra parte el driver ofrece la posibilidad de establecer otro tiempo mediante su parámetro DriverP3, en el que se le informa cuantos segundos debe esperar a que llegue un codigo. Por ejemplo, si se coloca el valor 7200, el objeto esperara 2 horas a que llegue un codigo antes de devolver a la aplicacion un error de timeout. Si se deja en 0, el objeto nunca saldra por error de timeout y se quedara eternamente esperando un codigo.

En resumen, al dispararse el driver, el objeto que lo maneja (HMITalk o NetTalk) quedará a la espera indefinidamente hasta que ocurra alguna de estas situaciones:

- Que entre una secuencia de caracteres y por un periodo de Timeout no hayan llegado mas caracteres, en cuyo caso se considerara un codigo completo.
- Que haya expirado el tiempo de espera maximo a que llegue un codigo definido en DriverP0.
- Que la aplicacion haya abortado la espera mediante la llamada al metodo de abortar la comunicacion.

En el caso que haya llegado un codigo, el objeto terminará su ejecucion y la aplicacion debera asegurarse de ponerlo nuevamente en modo escucha para que quede en condiciones de recibir un nuevo codigo.

Cuando llega un codigo y el driver esta siendo manejado por HMITalk, este emitira los eventos correspondientes. Se recomienda escuchar el evento OnSuccessfullyReceived ya que el OnPointValueChanged puede no resultar de utilidad. Para que el objeto quede escuchando nuevamente otro codigo, puede dejarse activada la propiedad ScanActive o bien llamar explicitamente al metodo Trigger una vez procesado el codigo recibido.

Si el driver esta siendo manejado desde NetTalk, este devolvera el control a la aplicacion donde se debera procesar el codigo recibido y luego volver a llamar al metodo ReadNumeric para ponerlo otra vez en modo escucha. Dado que NetTalk bloquea el flujo del programa cuando se lo utiliza en modo bloqueante, se recomienda manejarlo mediante un hilo secundario (por ejemplo con un control .NET para multitarea tipo BackgroundWorker).

SETTINGS:

Tanto el puerto virtual como el HMITalk o NetTalk que se utilicen se deben configurar con los siguientes settings:

9600,N,8,1

Si la base se conecta mediante un puerto serial, se puede utilizar un conversor/adaptador ethernet/serial para enlazarla a la PC a traves de la red LAN. En este caso, el adaptador es quien debera trabajar con los settings 9600,N,8,1 y los objetos HMITalk y NetTalk se deberan apuntar a una direccion IP en lugar de a un puerto serie.

CPKSoft Engineering
Industrial communication
drivers.

www.cpksoft.com
www.facebook.com/
cpksoftengineering
cpksoftengineering@
hotmail.com
phone: 54-911-45788354

1990-2015

Industrial communication solutions for Windows

Command list

Esperar Codigo

Description of this command:

Pone al driver a esperar un nuevo codigo transmitido desde el lector. Permite preformatear el codigo recibido y subdividirlo en subcampos de longitudes conocidas si se define un valor de DriverNumPoints mayor que 1 y se establece un formato en DriverP6. Los subcampos seran devueltos en tantos PointText como NumPoints se hayan reservado. Para cada subcampo, se devolvera a su vez en su PointValue correspondiente un valor numerico de entre 6 y 9 digitos siempre que sea posible obtenerlo, a partir del codigo recibido, y limitado al valor maximo que puede contener un punto flotante de 4 bytes (). Por lo general se devolvera el valor que sea posible formar con los primeros caracteres del codigo, hasta encontrar la primer letra en caso de tener texto. Debe tenerse en cuenta que por la limitacion en la cantidad de digitos significativos no es posible convertir a numero un codigo completo por ejemplo tipo EAN13, el que debera convertirse a numero directamente en la aplicacion a partir del PointText (donde si estara completo).

Methods used to run this command:

Analog Input (ReadNumericValues)

Number of points accepted by this command:

1-100

Meaning of the DriverP0 parameter:

Tiempo en milisegundos entre caracteres, pasado el cual se asume que se ha terminado de recibir un codigo si ya no entraron nuevos caracteres.

Meaning of the DriverP1 parameter:

Cantidad de caracteres esperados para el codigo. El maximo es 1000 caracteres. Si se deja en 0, la cantidad de caracteres es indefinida y no se aplica la deteccion lor longitud. Si se establece un valor, se devolvera el control a la aplicacion al recibir esa cantidad, incluso si todavia no se cumpliero el Timeout del puerto.

Meaning of the DriverP2 parameter:

Tiempo en segundos que se espera la llegada de un codigo completo, antes de devolver error de timeout a la aplicacion. Si se deja en 0, la espera es por tiempo indefinido.

Meaning of the DriverP6 parameter:

Lista separada por comas de longitudes de campos en los que se subdividira el codigo, siendo el primer numero la longitud que luego se volcara en PointText(0), y asi siguiendo. Si la cantidad de subcampos definidos en DriverNumPoints es menor que los definidos en P6 y sobran caracteres, el sobrante del codigo se pierde. Si la cantidad de subcampos definidos en DriverNumPoints es mayor que los definidos en P6 y sobran caracteres, el sobrante del codigo se asigna al primer PointText libre. Ejemplo: Suponiendo que DriverNumPoints se haya establecido a 4, la lista '2,4,4,6' devolvera los primeros 2 caracteres en PointText(0), los siguientes 4 caracteres en PointText(1), los siguientes 4 caracteres en PointText(2) y los siguientes 6 caracteres en PointText(3).

Values that are returned:

Value in PointValue (0) = Valor que se pudo obtener del codigo

Text in PointText (0) = Codigo alfanumerico recibido (hasta 1000 caracteres)

...

Value in PointValue (0 a NumPoints-1) = Si se separa en subcampos, valor que se pudo obtener del codigo para ese subcampo

Text in PointText (0 a NumPoints-1) = Si se separa en subcampos, codigo alfanumerico recibido (hasta 1000 caracteres) para ese subcampo

Error messages

The following list shows the possible error messages that can be returned by the driver during a failed communication in the 'Status' property.

[1005] DRIVER (Internal): Invalid driver stage
[1010] DRIVER (Internal): Error calculating elapsed milliseconds
[1201] DRIVER (System): Error closing %s
[1202] DRIVER (System): Error creating %s
[1208] DRIVER (System): Error seeking end of %s
[1210] DRIVER (System): Error writing to %s

Industrial communication solutions for Windows

[1214] DRIVER (System): Error deleting %s
[1300] PROTOCOL (Timeout): No answer
[1313] PROTOCOL (Timeout): No answer from meter after retrying with a Start Communications message
[1332] PROTOCOL (Remote): Invalid date received
[1333] PROTOCOL (Remote): Couldn't decode received date
[1334] PROTOCOL (Remote): Invalid time received
[1338] PROTOCOL (Remote): Couldn't decode reference date
[1345] PROTOCOL (Timeout): No barcode received after timeout period
[2001] CONFIG (DataType): Analog outputs are not supported by this driver
[2002] CONFIG (DataType): Digital inputs are not supported by this driver
[2003] CONFIG (DataType): Digital outputs are not supported by this driver

Supported devices

This driver can communicate with these devices, but is not necessarily limited to this list:

Serial Barcode Readers
Motorola LI-4278 BarCode Readers

CPKSoft Engineering

Industrial communication
drivers.

www.cpksoft.com

[www.facebook.com/
cpksoftengineering](http://www.facebook.com/cpksoftengineering)

[cpksoftengineering@
hotmail.com](mailto:cpksoftengineering@hotmail.com)

hotmail.com

phone: 54-911-45788354

1990-2015