**MANUAL DE INSTALACIÓN DEL SOFTWARE DE COMUNICACIÓN CON CONTROLADORES EPOS2 DESDE MATLAB**

1. **Introducción**

Este manual es para instalar y configurar el software de comunicación de los controladores de motores Maxon EPOS2 de manera que se puedan comandar desde Matlab y Simulink.

1. **Pre-requisitos**Los siguientes pre-requisitos de software son necesarios antes de poder usar la herramienta EPOS2.

* Matlab 2013b o superior.
* Compilador soportado por Matlab, para el caso de Matlab 2013b se requiere, para el caso de Windows el SDK y en Linux el compilador GCC.
* Driver USB de controlador EPOS2 instalado, tal como lo especifica Maxon Motors.
* Librerías (dll en Windows o .so para Linux) de Comunicación con los EPOS2 instaladas según las especificaciones de Maxon Motors.

1. **Instalación y validación de los pre-requisitos**
   1. **Validación del compilador**

**Para Windows:**

Para validar el compilador soportado por MATLAB, escriba en la ventana de comando la palabra ‘mex –setup’.

>> mex –setup

Ante esto, Matlab responde con una información relacionada con la búsqueda de los compiladores instalados en el sistema.

|  |
| --- |
| Welcome to mex -setup. This utility will help you set up  a default compiler. For a list of supported compilers, see  http://www.mathworks.com/support/compilers/R2013b/win64.html    Please choose your compiler for building MEX-files:    Would you like mex to locate installed compilers [y]/n? |

Teclee la letra ‘y’ para que empiece la búsqueda de compiladores. Matlab hará la búsqueda en el sistema y visualizará los compiladores detectados, pidiendo al usuario que seleccione el compilador a utilizar.

|  |
| --- |
| Select a compiler:  [1] Microsoft Visual C++ 2012 in C:\Program Files (x86)\Microsoft Visual Studio 11.0  [0] None |

En el caso ilustrado, el computador tiene instalado ‘Visual Studio C++ 2012’, ´por lo que aparece como opción única. En el caso de que no aparezca ninguna opción debe instalar uno de los compiladores soportado por Matlab. Para el caso de la versión 2013b, el listado de compiladores soportado se encuentra en la página,

<http://www.mathworks.com/support/compilers/R2013b/index.html?sec=win64>

Para el ejemplo ilustrado, seleccione la opción 1 y presione enter, con lo cual Matlab lanzará un mensaje de confirmación solicitando la validación de la información.

|  |
| --- |
| Please verify your choices:    Compiler: Microsoft Visual C++ 2012  Location: C:\Program Files (x86)\Microsoft Visual Studio 11.0    Are these correct [y]/n? |

A continuación acepte la opción de Matlab, con lo cual se da por finalizado la configuración del compilador a utilizarse.

**Para Linux:**

En el caso de Linux se debe instalar el compilador GCC previo a la configuración de Matlab. Este compilador es soportado por la mayoría de distribuiciones, por lo que no debe ser inconveniente su instalación.

Una vez instalado el compilador GCC, se procede a configurar el equipo de la misma forma que se hace para Windows, con el comando mex,

>> mex –setup

El mensaje de Matlab suele ser el mismo que en el caso de Windows, la diferencia es que ahora aparece el compilador GCC como el detectado en el sistema.

|  |
| --- |
| Select a compiler:  1: /opt/matlab2013b/bin/mexopts.sh  Template Options File for building gcc MEX - files  0: Exit with no change |

Ante lo cual se selecciona la opción 1 y Matlab procede a instalar el archivo de configuración de mex bajo Linux.

* 1. **Instalación del software de los EPOS2**

**Instalación Windows**

Para el caso de Windows, es necesario instalar el driver USB y las librerías o dll para la programación. Ambos se descargan desde la página web del producto. En la figura se ilustran ambos software.

|  |
| --- |
|  |

El driver USB, con nombre “EPOS USB Driver Installation” no requiere de pasos adicionales para su instalación, salvo descargarlo y ejecutarlo.

En el caso de las librerías .dll, se extrae el archivo que se descarga al hacer clic en “EPOS Windows 32-Bit and 64-Bit DLL”, en un directorio, para posteriormente copiar en una carpeta aparte los archivos con nombre: “Definitions.h”, “EposCmd.dll”, “EposCmd.lib”, “EposCmd64.dll”, “EposCmd64.lib”, “vxlapi.dll” y “vxlapi64.dll”. Esta carpeta es la que se utilizará durante la compilación de la librería de Matlab.

**Instalación Linux**

Para el caso de Linux, se procede a hacer clic en el enlace “EPOS Linux Library (ARMv6, ARMv7)”, el archivo se extrae en una carpeta y se procede a instalar en el sistema las librerías de enlace dinámico con terminación “.so”. Usualmente el directorio de instalación es la carpeta del sistema “/usr/lib” o “/usr/lib64”, pero esto depende de la distribución Linux.

A continuación se ilustra los comandos a seguir para el caso de un Linux de 64 bits con carpeta del sistema “/usr/lib”.

|  |
| --- |
| > mkdir epos2libs  > unzip EPOS-Linux-Library-En.zip –d ./epos2libs  > cd ./epos2libs/EPOS\_Linux\_Library/lib/x86\_64  > ln –s ./ libEposCmd.so.5.0.1.0 ./libEposCmd.so  > ln –s ./ libftd2xx.so.1.1.12 ./libftd2xx.so  > sudo mv \* /usr/lib |

Por último, para el caso de Linux, hay que instalar las reglas de “udev” que se encuentran en la carpeta “misc” de manera que se reconozca de manera automática el EPOS2 cuando se conecte por USB. Es importante agregar permisos para que los usuarios sin privilegios “root” puedan acceder al dispositivo, en caso contrario todas las aplicaciones que quieran hacer uso de la librería de los EPOS deben tener privilegios de administrador.

1. **Instalación del software**

Extraiga el archivo comprimido .zip en una carpeta, preferiblemente con nombre “Epos2SourceCode” al interior de la carpeta “MATLAB”. En esa carpeta se encontrará el siguiente listado de archivos.

|  |  |
| --- | --- |
| **Nombre** | **Descripción** |
| *Epos2Linux* | Carpeta donde alojar los archivos de Linux |
| *Epos2Windows* | Carpeta donde alojar los archivos de Windows |
| Clean | Archivo para borrar los archivos de instalación |
| ClearErrorState.c | Código fuente para función mex |
| ClearErrorState.m | Ayuda de Matlab para la función mex |
| CloseCommunication.c | Código fuente para función mex |
| CloseCommunication.m | Ayuda de Matlab para la función mex |
| DisableNode.c | Código fuente para función mex |
| DisableNode.m | Ayuda de Matlab para la función mex |
| EnableNode.c | Código fuente para función mex |
| EnableNode.m | Ayuda de Matlab para la función mex |
| Epos2.m | Definición de la clase EPOS2 |
| FindHome.c | Código fuente para función mex |
| FindHome.m | Ayuda de Matlab para la función mex |
| GetCurrent.c | Código fuente para función mex |
| GetCurrent.m | Ayuda de Matlab para la función mex |
| GetErrorState.c | Código fuente para función mex |
| GetErrorState.m | Ayuda de Matlab para la función mex |
| GetObject.c | Código fuente para función mex |
| GetObject.m | Ayuda de Matlab para la función mex |
| GetOperationMode.c | Código fuente para función mex |
| GetOperationMode.m | Ayuda de Matlab para la función mex |
| GetPosition.c | Código fuente para función mex |
| GetPosition.m | Ayuda de Matlab para la función mex |
| GetVelocity.c | Código fuente para función mex |
| GetVelocity.m | Ayuda de Matlab para la función mex |
| HommingMethods.m | Ayuda de Matlab para la función mex |
| IsTargetReached.c | Código fuente para función mex |
| IsTargetReached.m | Ayuda de Matlab para la función mex |
| Make.m | Archivo instalador de la librería |
| MoveToPosition.c | Código fuente para función mex |
| MoveToPosition.m | Ayuda de Matlab para la función mex |
| MoveWithVelocity.c | Código fuente para función mex |
| MoveWithVelocity.m | Ayuda de Matlab para la función mex |
| OpenCommunication.c | Código fuente para función mex |
| OpenCommunication.m | Ayuda de Matlab para la función mex |
| OperationModes.m | Ayuda de Matlab para la función mex |
| **Nombre** | **Descripción** |
| QuickStop.c | Código fuente para función mex |
| QuickStop.m | Ayuda de Matlab para la función mex |
| SetCurrent.c | Código fuente para función mex |
| SetCurrent.m | Ayuda de Matlab para la función mex |
| SetObject.c | Código fuente para función mex |
| SetObject.m | Ayuda de Matlab para la función mex |
| SetOperationMode.c | Código fuente para función mex |
| SetOperationMode.m | Ayuda de Matlab para la función mex |
| SetPosition.c | Código fuente para función mex |
| SetPosition.m | Ayuda de Matlab para la función mex |
| SetProfilePositionData.c | Código fuente para función mex |
| SetProfilePositionData.m | Ayuda de Matlab para la función mex |
| SetProfileVelocityData.c | Código fuente para función mex |
| SetProfileVelocityData.m | Ayuda de Matlab para la función mex |
| SetVelocity.c | Código fuente para función mex |
| SetVelocity.m | Ayuda de Matlab para la función mex |
| WaitForTargetReached.c | Código fuente para función mex |
| WaitForTargetReached.m | Ayuda de Matlab para la función mex |
| sfun\_maxon.c | Código fuente de la s-function a usarse con Simulink |
| testMaxonMotors.slx | Ejemplo de uso de la s-function en Simulink |
| test\_epos2.m | Ejemplo de uso de las funciones mex |
| waitfortreal.m | Soft-Real Time con Simulink |

Copie el archivo “Definitions.h” que viene con las librerías de los EPOS2 en la carpeta “Epos2Linux” o “Epos2Windows” según la arquitectura de trabajo del computador.

Abra MATLAB y cambie el directorio de trabajo a la carpeta donde se encuentran los archivos extraídos.

Ahora ejecute el comando “Make”.

>> Make

Si todo se encuentra correctamente se visualizará los siguientes mensajes.

|  |
| --- |
| Welcome to EPOS2 library  Eugenio Yime, 2015  .. Compiling for a Windows computer  .. finishing compiling  .. adding directory to path  .. done, you can now use this library  Please, use the command: >> savepath  if you want to permanently add the source code directory to Matlab |

Es importante tener en cuenta que el archivo “Make” no agrega de forma permanente el directorio de compilación a la variable “PATH” de Matlab. Si el usuario quiere hacer uso de la librería de manera permanente, debe utilizar el comando “savepath”.