Dobot Magician Demo Description (MATLAB)

Issue: V1.0
Date: 2018-06-30

Shenzhen Yuejiang Technology Co., Ltd
Copyright © Shenzhen Yuejiang Technology Co., Ltd 2018. All rights reserved.

No part of this document may be reproduced or transmitted in any form or by any means without prior written consent of Yuejiang Technology Co., Ltd

Disclaimer

To the maximum extent permitted by applicable law, the products described (including its hardware, software and firmware, etc.) in this document are provided AS IS, which may have flaws, errors or faults. Yuejiang makes no warranties of any kind, express or implied, including but not limited to, merchantability, satisfaction of quality, fitness for a particular purpose and non-infringement of third party rights. In no event will Yuejiang be liable for any special, incidental, consequential or indirect damages resulting from the use of our products and documents.

Before using our product, please thoroughly read and understand the contents of this document and related technical documents that are published online, to ensure that the robotic arm is used on the premise of fully understanding the robotic arm and related knowledge. Please use this document with technical guidance from professionals. Even if follow this document or any other related instructions, Damages or losses will be happen in the using process, Dobot shall not be considered as a guarantee regarding to all security information contained in this document.

The user has the responsibility to make sure following the relevant practical laws and regulations of the country, in order that there is no significant danger in the use of the robotic arm.

Shenzhen Yuejiang Technology Co., Ltd

Address: 4F, A8, Tanglang Industrial Area, Taoyuan Street, Nanshan District, Shenzhen, PRC
Website: www.dobot.cc
Preface

Change History

<table>
<thead>
<tr>
<th>Date</th>
<th>Change Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2018/06/30</td>
<td>The first release.</td>
</tr>
</tbody>
</table>

Symbol Conventions

The symbols that may be founded in this document are defined as follows.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>🔄 DANGER</td>
<td>Indicates a hazard with a high level of risk which, if not avoided, could result in death or serious injury</td>
</tr>
<tr>
<td>🔄 WARNING</td>
<td>Indicates a hazard with a medium level or low level of risk which, if not avoided, could result in minor or moderate injury, robotic arm damage</td>
</tr>
<tr>
<td>🔄 NOTICE</td>
<td>Indicates a potentially hazardous situation which, if not avoided, can result in robotic arm damage, data loss, or unanticipated result</td>
</tr>
<tr>
<td>🔄 NOTE</td>
<td>Provides additional information to emphasize or supplement important points in the main text</td>
</tr>
</tbody>
</table>
## Contents

1. MATLAB Demo ........................................................................................................... 1
   1.1 Environment Building.......................................................................................... 1
   1.2 MATLAB Demo Description.................................................................................. 3
       1.2.1 Project Description....................................................................................... 3
       1.2.2 Code Description......................................................................................... 3
1. **MATLAB Demo**

This topic is aimed at helping user to understand common API of Dobot Magician and build development environment quickly.

1.1 **Environment Building**

This demo is developed with **MATLAB** and compiled with **TDM- GCC**. You need to install MATLAB and TDM-GCC with 64-bits. The download path of TDM-GCC is [http://tdm-gcc.tdragon.net/download](http://tdm-gcc.tdragon.net/download).

This topic takes **Windows 10** OS as an example to describe how to install and configure MATLAB. Please replace it based on site requirements.

**Procedure**

**Step 1**  Take MATLAB 2016Ra as an example, install MATLAB. The details how to install is not described in this topic.

**Step 2**  Install TDM-GCC. The details how to install is not described in this topic.

⚠️ **NOTICE**

The installation paths of MATLAB and TDM-GCC cannot contain Chinese character and space.

**Step 3**  Create system variable **MW_MINGW64_LOC** and set to the installation path of TDM-GCC, as shown in Figure 1.1.

If the installation path of TDM-GCC is **C:\TDM-GCC-64**. Please replace it based on site requirements.

![Figure 1.1 Add system variable](image)

**Step 4**  Run `setenv('MW_MINGW64_LOC','C:\TDM-GCC-64')` command in the command window of MATLAB, as shown in Figure 1.2
Step 5  Add the directory of Dobot DLL to the search path of MATLAB, as shown in Figure 1.3.

Step 6  Restart MATLAB and open MATLAB demo.

Step 7  After Dobot Magician is connected to PC and powered on, you can click on the MATLAB page.
1.2 MATLAB Demo Description

1.2.1 Project Description

The **loadDLL.m** file is the Run file of MATLAB demo, which shows how to load Dobot DLL and call the Dobot Magician APIs.

After loading Dobot DLL, you can run **libfunctions DobotDll -full** command in the command window of MATLAB to view all Dobot Magician API declarations. For details about API description, please see **Dobot Magician API Description**.

1.2.2 Code Description

In this demo, we use command queue mode. Figure 1.5 shows the realization process of the MATLAB demo.
Figure 1.5  Realization process

(1)  Load Dobot DLL.

Program 1.1  Load Dobot DLL

loadlibrary('DobotDll.dll','DobotDll.h');

(2)  Connect to Dobot Magician.

Program 1.2  Connect to Dobot Magician

%create pointer piont to charArrays
str1= libpointer('cstring',ch);
%search the magician device   res1:device number   res2:device address
[res1,res2]=calllib('DobotDll','SearchDobot',str1,128);

%create pointer piont to device address
str2= libpointer('cstring',res2);

%contect device res3:contect result 0:success 1:error 2:timeout
[res3,res4]=calllib('DobotDll','ConnectDobot',str2, 115200);
(3) Start executing command queue.

Program 1.3 Start executing command queue

```matlab
% cmd start exec queue
calllib('DobotDll','SetQueuedCmdStartExec')
```

(4) Run PTP command.

Program 1.4 Run PTP command

```matlab
% create c type struct
ptpstruct=libstruct('tagPTPCmd',ptp);

% create ptp pointer
ptpstruct_ptr=libpointer('tagPTPCmdPtr',ptpstruct);
queue_index_ptr=libpointer('uint64Ptr',queue_index);

% send ptp cmd
calllib('DobotDll','SetPTPCmd',ptpstruct_ptr,true,queue_index_ptr);
```

(5) Stop executing command queue.

Program 1.5 Stop executing command queue

```matlab
% cmd stop exec queue
calllib('DobotDll','SetQueuedCmdStopExec');
```

(6) Disconnect Dobot Magician.

Program 1.6 Disconnect Dobot Magician

```matlab
% cmd disconnect device
calllib('DobotDll','DisconnectDobot');
```