

Dynamixel MX-28 Motor Library

Table of Contents

1. Libraries Overview

1.1. Introduction

1.2. Using the Library

2. Functions

2.1. M28_lib.h

2.1.1. Factory reset

2.1.2. Present Temperature

2.1.3. Present Voltage

2.1.4. Change ID

2.1.5. Position

2.1.6. Load Check

2.1.7. Present Speed

2.2. Uartmsp.h

2.2.1. UART Configuration

2.2.2. Send data

2.2.3. Receive data

2.2.4. Check Sum

3. Recommended Read

Library Overview

1.1. INTRODUCTION

The purpose of this document is to help to use and understand the operation of the Dynamixel MX28 Motor with the described library. The related functions are:

- I. UART Configuration
- II. Factory Reset
- III. Present Temperature
- IV. Present Voltage
- V. Change ID
- VI. Position
- VII. Present Position
- VIII. Present Load

In addition, the `#defines` used in these functions will also be explained.

1.2. Using the Library

Building an application which uses the Dynamixel MX28 Motor Libraries will require the include of two files: `m28_lib.h` and `uartmsp.h`.

The `m28_lib.h` is a header file which provides all the functions and its `#defines`, except the UART Configuration one that is in the `uartmsp.h` with its defines. The functions itself were built in the `.c` correspondent file on each `.h` file.

Functions

2.1. M28_lib.h

2.1.1. Factory Reset

DESCRIPTION

This function resets the Control Table of MX28 Motor to its factory default settings.

SYNTAX

INPUT: Equivalent Motor ID
`factory_reset(id);`

2.1.2. Present Temperature

DESCRIPTION

This function reads the internal temperature of the Motor. Data value is identical to the actual temperature in Celsius.

SYNTAX

INPUT: Equivalent Motor ID
`temp_check(id);`

RETURN

Returns the 5th element of a storage vector that is the temperature value sent by the motor.

```
temp_out = temp_rx[5];
```

2.1.3 Present Voltage

DESCRIPTION

This function reads the size of the current supplied voltage. The value in the vector is 10 times larger than the actual voltage.

SYNTAX

INPUT: Equivalent Motor ID
`voltage_check(id);`

RETURN

Returns the 5th element of a storage vector divided by 10, that is the actual supplied voltage of the motor if the data value is correct. If it is not, returns 0.

```
voltage_out = voltage_rx[5]/10;
```

2.1.4 Change ID

DESCRIPTION

This function changes the current Motor ID. The range from 0 to 252 can be used, and, especially, 254 is used as the Broadcast ID.

SYNTAX

INPUT: Current Motor ID and the next ID
`change_id(id,newid);`

2.1.5. Position

DESCRIPTION

This function reads the current position of the motor access in degrees and is divided in two instructions: One for read the lowest significative byte and other to read the highest significative byte. After this, the two instructions mix this two 8 bits informations in one information of 16 bits.

SYNTAX

INPUT: Equivalent Motor ID
`position_check(id);`

RETURN

Returns two 8 bits informations: one of the lowest significative bytes, and the other of the highest significative bytes. The highest significative byte is displaced in 8 bits, than mixed with the lowest significative byte composing the value of position, that is multiplied by 0.088.

```
position_h = position_h_rx[5] << 8;  
position_l = position_l_rx[5];  
position_out = position_h + position_l;  
position_out = position_out * 0.088;
```

2.1.6. Load Check

DESCRIPTION

Reads the current applied load. The range of the value is 0 to 2047. If the value is in a range of 0 to 1023, it means the load works in CCW direction. If the value is in a range of 1024 to 2047 then the motor rotates in CW direction.

SYNTAX

INPUT: Equivalent Motor ID
`load_check(id);`

RETURN

If the load is in CCW direction:

```
load_total= (parameter/1023)*100;  
return 1;
```

If is in CW direction:

```
load_total = (parameter-1023);  
load_total = load_total/1023;
```

2.1.7. Present Speed

DESCRIPTION

Reads the current motor axcess moving speed. The range of the value is 0 to 2047. If the value is in a range of 0 to 1023, it means the motor rotates in CCW direction. If the value is in a range of 1024 to 2047 then the motor rotates in CW direction.

SYNTAX

INPUT: Equivalent Motor ID
`speed_check(id);`

RETURN

Returns two 8 bits informations: one of the lowest significative bytes, and the other of the highest significative bytes. The highest significative byte is displaced in 8 bits, than mixed with the lowest significative byte composing the value of speed.

```
speed_h = speed_h_rx[5] << 8;
speed_l = speed_l_rx[5];
speed_param = speed_h + speed_l;

if( speed_param <= 1023){
    speed_out= speed_param*0.11;
    speed_direction = 1

else if(1023 < speed_param <=2047)
    speed_out= ((speed_param-1023)*0.11;
    speed_direction=0;
```


2.2 Uartmsp.h

2.2.1. Uart Configuration

DESCRIPTION

Configure the microcontrolled unit to allow the communication with the MX28 Motor.

SYNTAXE

```
uart_config(void);
```

2.2.2 . Send data

DESCRIPTION

Function to send package.

SYNTAXE

```
send_data(tx_vector, size);
```

2.2.3. Receive data

DESCRIPTION

Function to receive package from the MX28.

SYNTAXE

```
void receive_data(rx_vector,size);
```

3. Recommended Read

http://support.robotis.com/en/product/dynamixel/mx_series/mx-28.htm#Actuator_Address_26

http://support.robotis.com/en/product/dynamixel/communication/dxl_packet.htm

http://support.robotis.com/en/product/dynamixel/communication/dxl_instruction.htm