Quick Start for DXL-IO

IO Module for Dynamixel
ADC/ Digital I/O / PWM

Introduction

This document describes the functionality of DXL-IO module which includes 8 channel of Analog to Digital Convertor (ADC), 8 channel of Digital I/O pin and 8 channel of Pulse With Modulation (PWM) available for controllers using Dynamixel Bus. This module is desinged as a cheap and lightweight Input/output module to be used with devices using TTL Dynamixel communication protocol. The module can drive servo motors, read analog sensors drive with internal filtering, digital input and output with internal AVR controller which can provide high speed integration. The DXL-IO can be controlled via a serial interface and therefore which are simple and compact enough to be further processed by a low power CPU like a microcontroller using ROBOTIS Dynamixel TTL communication protocol.
Users who are not familiar with the **ROBOTIS DYNAMIXEL** TTL communication protocol can obtain additional information from the documents published by ROBOTIS and AX-12 Datasheet.

Module Specifications

- Small, Low weight: approx. 15g .
- Supports communication baud rate of **1Mbps (fixed)**.
- ROBOTIS TTL communication protocol facilitates direct attachment of the device to a Dynamixel communication bus.
- **8 pin Input / Output** in digital level (0~5 volt input and output) or **8 channel Servo** motor drive.
- **8 pin ADC (8bit and 10bit)**
- Easy communication and provide external power for external sensors power.
- Status controllable LED onboard.
- Internal ADC low Pass filter

Device Operation

The DXL-IO can be operates with one external main controller (like CM-5 or CM-700 from Robotis) or handmade controller like AVR/ARM microcontrollers or PC with USB2Dynamixel or USBzDXL from ***PishRobot.Co***. Module settings like Id will be saved into the FLASH/EEPROM memory of the module (**default ID is 100 (0xff)**). After all, the module is only connected to the CM5 and can receive commands to control and process the commands and also return requested data from ADC and I/O pins or can set servo motors.

ADC Mode:

This mode is always available for **8 channel** form **J9 to J16** connectors. for **8bit** use you can use address **78** to **85**, and also for **10bit** resolution you can use address **[24,25] for first channel, [26,27] for second channel** and etc. according to register table in **10bit** resolution first address is low byte and second one is high byte of data.

SERVO Motor Mode:

For controlling Servo motors from I/O pins, you can just connect it to **J1** to **J8** port and write registers **60** to **67** from value **1** to **20** for position control. When you want to use one port as I/O port, its address must be set 0.

I/O Mode:

This mode is used for input/output digital mode on **J1** to **J8** ports. When you want to read pins status just read it's register (**address 90 to 97**), module internally change pin to **input mode**. When user want to use this ports as a output, just write data on their registers address from **90** to **97**, and controller change port to **output mode**.

Register Description

Address	Register	Access	Default Value	Range Value
0	Model Number-L	RO	?	NA
1	Model Number-H	RO	?	NA
2	Version of firmware	RO	?	NA
3	Id	RD/WR	100	0~253
4	Baud Rate	RO	1	Fix
5	Return Time	RD/WR	0	0~255
6	Status Return Level	RD/WR	1	0~2
7	Reserved	-	-	-
8	Filter Precent (ADC filter)	RD/WR	10	0~255
16	LED (1Byte)	RD/WR	2	0~2
24	ADC(0) L	RD	?	-
25	ADC(0) H	RD	?	-
26	ADC(1) L	RD	?	-
27	ADC(1) H	RD	?	-
28	ADC(2) L	RD	?	-
29	ADC(2) H	RD	?	-
30	ADC(3) L	RD	?	-
31	ADC(3) H	RD	?	-
32	ADC(4) L	RD	?	-
33	ADC(4) H	RD	?	-
34	ADC(5) L	RD	?	-
35	ADC(5) H	RD	?	-
36	ADC(6) L	RD	?	-
37	ADC(6) H	RD	?	-
38 or 48	ADC(7) L	RD	?	-
39 or 49	ADC(7) H	RD	?	-
78	8bit ADC(0)	RD	?	-
79	8bit ADC(1)	RD	?	-
80	8bit ADC(2)	RD	?	-
81	8bit ADC(3)	RD	?	-
82	8bit ADC(4)	RD	?	-
83	8bit ADC(5)	RD	?	-
84	8bit ADC(6)	RD	?	-
85	8bit ADC(7)	RD	?	-
60	Servo(0)	WR	0	0~20

61	Servo(1)	WR	0	0~20
62	Servo(2)	WR	0	0~20
63	Servo(3)	WR	0	0~20
64	Servo(4)	WR	0	0~20
65	Servo(5)	WR	0	0~20
66	Servo(6)	WR	0	0~20
67	Servo(7)	WR	0	0~20
90	I/O(0)	RD/WR	?	0~1
91	I/O(1)	RD/WR	?	0~1
92	I/O(2)	RD/WR	?	0~1
93	I/O(3)	RD/WR	?	0~1
94	I/O(4)	RD/WR	?	0~1
95	I/O(5)	RD/WR	?	0~1
96	I/O(6)	RD/WR	?	0~1
97	I/O(7)	RD/WR	?	0~1

RO: These registers are read only, and cannot be written.

Important note

Based on the hardware and firmware design of the module, some of the registers cannot be changed or should only be changed in a limited scale; otherwise it leads to corrupted communication between the module and the main controller, or timing failures.

Initial Value

The Initial Value column on the right side of the control table shows the factory default values for the case of EEPROM area data, and shows the initial value when the power is turned on for the case of RAM area data. The following explains the meaning of data stored in each of the addresses in the control table.

Address 0x00, 0x01

Model Number.
For DXL-IO, this value is (14). For ver1.0.

Address 0x02

Firmware Version.
Version firmware of board. This document is proposed for ver.1.2. in Nov-2014.

Address 0x03

ID.
The unique Id number assigned to each module for identifying them. Different Ids are required for each module that are on the same network. **Default id** for this devise originally set to **100**.

Address 16

LED.
The LED turns on when set to 1 and turns off if set to 0. Default value is 2, this mean is show data exchange for communication.

Controlling

The communication between the module and CM5 or etc. has the same form as DYNAMIXEL servos such as AX-12. Therefore, users who are not familiar with this protocol are recommended to refer to AX-12 datasheet for more information. The next table shows the instructions available in the DXL-IO module.

Instruction	Function	Value	Number of Parameter
PING	No action. Used for Obtaining a status Pocket	0x01	0
READ DATA	Reading Values in the Control Table	0x02	2
WRITE DATA	Writing Values to the Control Table	0x03	2~3

-PING

This instruction is used to check whether the device exists and is ready to receive the next instruction. The instruction is the same as in AX-12.

-READ DATA

This instruction is used to read Present Value of Module. It is the same as the READ instruction in DYNAMIXEL.

-WRITE DATA

This instruction is to write the content of DXL-IO registers. It is the same as the WRITE instruction in Dynamixel except that it accepts only single or Twice byte write.

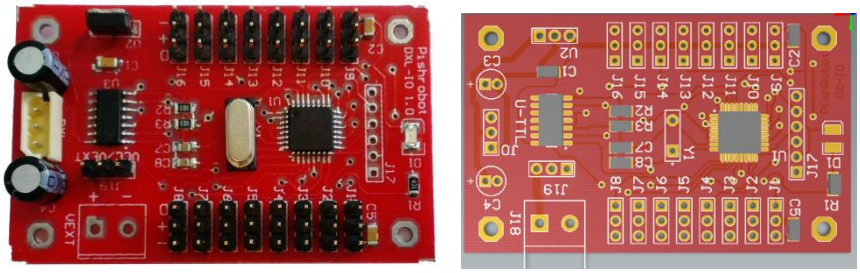

Fig.01 – DXL-IO Connection Pins.