



190q-6b-SZ-485-Motor-Control-HL-YZ- 2023-3-14.odp

CTI One Corporation
This document is created by:

Harry Li, Ph.D.
Youran Zheng
Yusuke Yakuwa

March 14, 2023

Motor Controller Test Procedures (1/29,24)

A. Test 485 software for the motor drive on the Windows Machine:

B. Test Python Code for the motor drive on the Jetson NANO

A1. Get the software on the Windows Machine

B1. Have the J41 on Jetson NANO cable ready to connect to the prototype board

B2. Add the level shifter for GPIO and PWM output

A2. Have the motor cable connection to the motor drive

B3. have UART output from J41 connected to the motor controller to establish 485 communication



Ref SZ Motor Controllers with 8 Documents (6/28-4/21, 2023)

CANopen standardizes communication between devices and applications from different manufacturers. It has been used in automation and motion applications

The Basics of CANopen Updated Mar 31, 2023

<https://www.ni.com/en-us/shop/seamlessly-connect-to-third-party-devices-and-supervisory-system/the-basics-of-canopen.html>



- 1How to connect to PC and use ZLAC8015(D) software.pdf
- 2ZLAC8015 CANopen Communication Routine V1.05.pdf
- 3ZLAC8015 MANUAL V2.00.pdf
- 4ZLAC8015 RS485 COMMUNICATION Version 1.03.pdf
- 5ZLAC8015_V1.0(2).eds
- 6ZLAC8015D_接线 (英文) .mp4
- 7ZLTECH-HMD-MAX-V2_0.exe
- 8ZLTECH-HMD-MAX-V2_0 使用说明 .pdf

/media/harry/easystore3/backup-2020-2-15/CTI/3projects/3-8-smart-tech/3-8-4-CTI/3-8-4-6-products/AIV200/190-robots-health/190q-manufacturing-production/190q-6-hub-motors/190q-6b-hub-motors-controllers/sz-motor-and-controller/ZLAC8015D\$

1. CAN and/or 485 interface for the motor controller



2. HL: CAN protocol over 485 hardware

From chatGPT: the CAN (Controller Area Network) protocol can be implemented on hardware that uses the RS-485 physical layer. While CAN and RS-485 are different protocols, they can be used together by incorporating a **CAN transceiver** that supports RS-485 signaling.

The RS-485 standard defines the electrical characteristics and signaling for communication over a differential pair of wires. On the other hand, CAN is a higher-level protocol that provides a robust and efficient communication method for devices on a network.

To implement CAN on RS-485 hardware, you would need a **CAN controller** and a **CAN transceiver that supports RS-485 signaling**. The CAN controller manages the CAN protocol and handles tasks such as message framing, error checking, and arbitration. The CAN transceiver ...

To use CAN over RS-485, you would need a transceiver that supports both protocols. This transceiver would convert the differential CAN signals to the RS-485 electrical signaling and vice versa. Additionally, you would require appropriate software or firmware to implement the CAN protocol on top of the RS-485 physical layer.

It's worth noting that while CAN and RS-485 share some similarities in their electrical signaling, they are different protocols with different features and capabilities. Adapting CAN to work over RS-485 may require additional considerations and modifications to ensure proper functionality and compatibility.

YK Motor Testing with SZ Motor Controller (1/22, 2024)

Notes: HL 2024-1-22

Jan 2nd Monday

Y

1. Motor Add. Angular Displacement
Sensor, Optical Encoder.

Two Wires (Differential Signal)

$$\frac{360}{1024} \approx 0.36 \text{ degrees. ... (1)}$$

1024 Measurement for 360° circle.

Note: Stepper Motor

$$\frac{360}{200 \text{ steps}} = 1.8^\circ/\text{step} \quad \dots (2)$$

$0,9^\circ$ Degree/Half Step ... (3)

$0,45^\circ$ Degree/ $\frac{1}{2}$ Step ... (4)

$0,225^\circ$ Degree/ $\frac{1}{8}$ Step ... (5)

Equation (1) with DTC Encoder is
Comparable with $0,9^\circ$ Step.

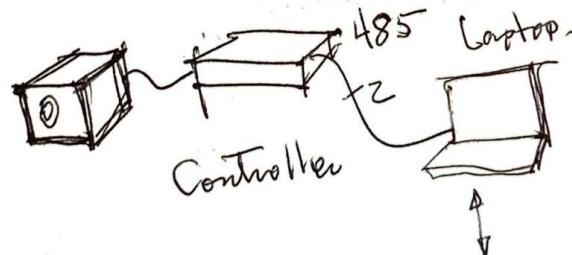
2. Wire Configuration of the motor
with Optic Encoder.

- { 5 Wires - Hall Sensor
 - 3 Wires - motor U, V, W
 - 2 Wires - Optical Encoder
 - 2 Wires - Thermal Sensor (Temperature Sensor)
- 12 Wires.

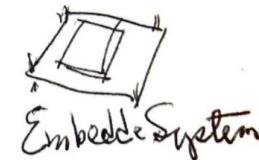
3. Torque Compensation Using
SZ motor Company's Controller

Load Fraction \rightarrow Torque Current
Increases \uparrow Increases By
motor controller \uparrow

bit Rate of the
testing over 485: 400b
Speed Kept
Constant.



Note: Python Code
Based Testing.



Python Code Testing of SZ Motor Controller (10/8-7/27,2023)

To prepare the python program test, make sure (1) to have the python code ready, from here:

```
(pymodbus) harry@harrys-gpu-laptop:~/PycharmProjects/w100/485-controller$ tree -L 1
.
└── 190q-6a-4-#485pymodbus39-2023-7-13.yml
    ├── main.py
    ├── modbus_test.py
    └── pymodbus.yml
```

(2) locate the pymodbus.yml file then create anaconda environment, using the following command:

```
$ conda env create -f pymodbus.yml
# Activate the Anaconda environment:
$ conda activate pymodbus
```

Note 2023-10-8 for installation of anaconda environment, see readme document by YZ, 190q-6a-1-README-pymodbus-YZ-2023-7-11 (1).txt

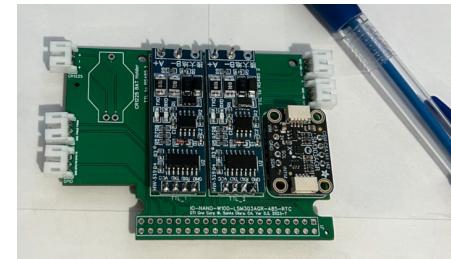
(3) once the pymodbus environment is created, then run the python testing code. If the GPIO port is connect connected, then you will see port not connected error message.

(4) both motor uses J41-8/10 with SZ controller, UART to 485 board;

UART1(left)_1	VCC	J41-1	3V3
UART1(left)_2	TXD	J41-10	UART_2_RX
UART1(left)_3	RXD	J41-8	UART_2_TX
UART1(left)_4	GND	J41-8	gnd
UART2(left)_1	VCC	J41-1	3V3
UART2(left)_2	TXD	J50_RX	J50-3
UART2(left)_3	RXD	J50_TX	J50-4
UART2(left)_4	GND	J41-8	gnd

Connect Tx/Rx to SZ controller to test motor control, perform loopback test (optional) if needed. Run minicom to set up the UART communication parameters by HL. Putty testing and Python code testing by Youran Zheng

Note 2023-10-8: SZ motor drive just need one UART.

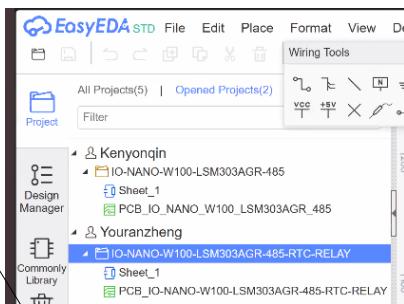
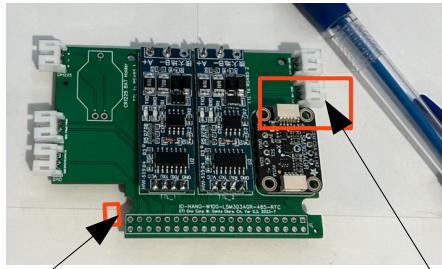


UART/485 Connection To Motor Controller (10/8-7/27,2023)

Continued from the previous slides

1. The CTI One's IO board UART to 485 conversion board:
input connectors (JST 3 pins connector) from NANO J41-8/J41-10
and GND;

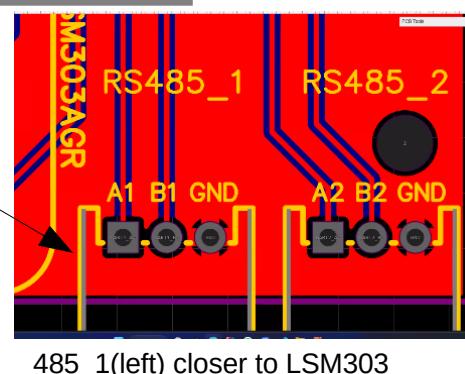
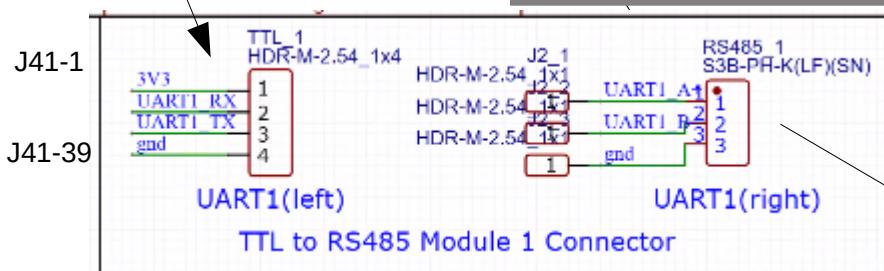
Note 2023-10-8: the pin connectivity is given in the Schematic from EasyEDA below



1
J41-8 Rx, J41-10 Tx

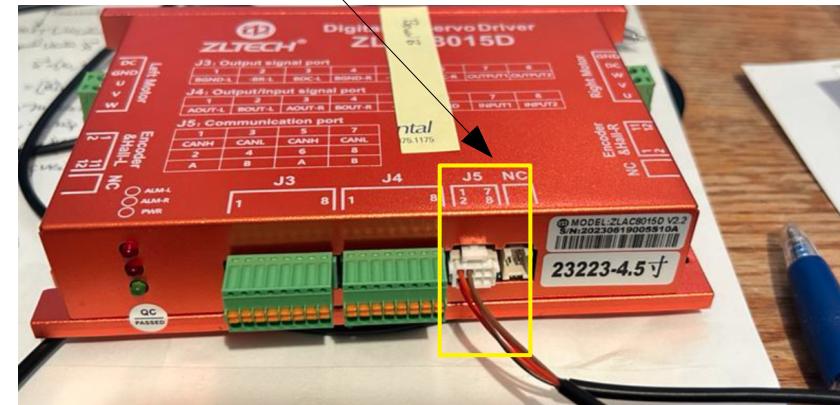
Note: connector for mounting 485 board

Note 2023-10-8: To save time for soldering location identification, screen print pin1, 8,10,39



2. 485 output from CTI One's IO board as the input to the SZ motor controller

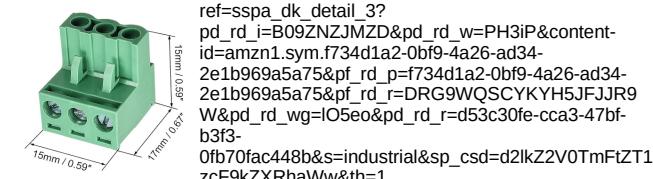
J5-1 A Orange (red)
J5-7 B Brown



20 Sets Mini Micro Jst 2.0 Ph 3 Pin Connector Plug Male With 150mm Cable & Female by daier, 8p99

[https://www.amazon.com/Micro-Connector-150mm-Cable-Female/dp/B01DUC1PW6/ref=sr_1_4?](https://www.amazon.com/Micro-Connector-150mm-Cable-Female/dp/B01DUC1PW6/ref=sr_1_4?crid=26S9FUJ62CBL&keywords=jst+connector+3+pin+male&qid=1696809717&sprefix=jst+connector+3+pin+male%2Caps%2C125&sr=8-4)

https://www.amazon.com/dp/B09NZJMZD/ref=sspa_dk_detail_3?pd_rd_i=B09NZJMZD&pd_rd_w=PH3iP&content_id=amzn1.sym.f734d1a2-0bf9-4a26-ad34-2e1b969a5a75&pf_rd_p=f734d1a2-0bf9-4a26-ad34-2e1b969a5a75&pf_rd_r=DRG9WQSCYKYH5FJJR9W&pd_rd_wg=I05eo&pd_rd_r=d53c30fe-cca3-47bf-b3f3-0fb70fac448b&s=industrial&sp_csd=d2lkZ2V0TmFtZT1zcF9kZXRaWw&th=1



Testing SZ Motor Controller (9/29-7/27,2023)

To prepare the python program test, make sure (1) to have the python code ready, from here:

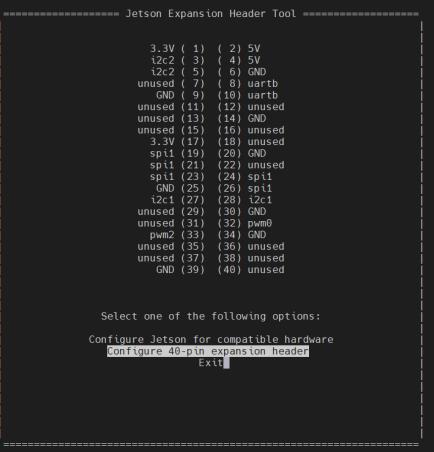
```
(pymodbus) harry@harrys-gpu-laptop:~/PycharmProjects/w100/485-controller$ tree -L 1
.
└── 190q-6a-4-#485pymodbus39-2023-7-13.yml
    ├── main.py
    ├── modbus_test.py
    └── pymodbus.yml
```

(2) locate the pymodbus.yml file then create anaconda environment, using the following command:

```
$ conda env create -f pymodbus.yml
# Activate the Anaconda environment:
$ conda activate pymodbus
```

(3) once the pymodbus environment is created, then run the python testing code.

Update on J50 UART Port (7/27,2023)



The screenshot shows two windows of the Jetson Expansion Header Tool. The left window lists pins from 3.3V to GND with their corresponding pin numbers. The right window shows a list of functions for pins, with 'pwm0' selected. Both windows have a 'Back' button at the bottom.

```

----- Jetson Expansion Header Tool -----
3.3V ( 1 ) ( 2 ) 5V
I2C2 ( 3 ) ( 4 ) 5V
I2C2 ( 5 ) ( 6 ) GND
unused ( 7 ) ( 8 ) uartb
GND ( 9 ) ( 10 ) uartb
unused (11) (12) unused
unused (13) (14) GND
unused (15) (16) unused
3.3V (17) (18) unused
spit (19) (20) GND
spit (21) (22) unused
spit (23) (24) spit
GND (25) (26) spit
I2C1 (27) (28) I2C1
unused (29) (30) GND
unused (31) (32) pwm0
pwm2 (33) (34) GND
unused (35) (36) unused
unused (37) (38) unused
GND (39) (40) unused

Select one of the following options:
Configure Jetson for compatible hardware
Configure 40-pin expansion header
Exit
  
```

----- Jetson Expansion Header Tool -----
Select desired functions (for pins):
 aud_mclk (7)
 I2C0 (12,35,38,40)
 pwm0 (32)
 pwr2 (33)
 spit (19,21,23,24,26)
 spit2 (13,16,18,22,37)
 uartb-cts/rts (11,36)

Back

----- extlinux.conf /boot/extlinux -----
Open Save & x
1 TIMEOUT 30
2 DEFAULT JetsonIO
3
4 MENU TITLE L4T boot options
5
6 LABEL primary
7 MENU LABEL primary kernel
8 LINUX /boot/Image
9 INITRD /boot/initrd
10 APPEND \${cbootargs} quiet root=/dev/mmcblk0p1 rw rootwait rootfstype=ext4 console=tty0 fbcon=map:0 net.ifnames=0
11 APPEND tegraid=21.1.2.0.0 ddr_die=4096M@2048M section=512M memtype=0 vpr_resize usb_port_owner_info=0 lane_owner_info=0
 emc_max_dvfs=0 touch_id=0@63 video=tegrafb no_console_suspend=1 debug_uartport=lsport,4 earlyprintk=uart8250-32bit,0x70006000 maxcpus=4
 usbcore.old_scheme_first=1 lp0_vec=0x1000@0xffff780000 core_edp_mv=1125 core_edp_ma=4000 gpt earlycon=uart8250,mmio32,0x70006000 root=/dev/mmcblk0p1 rw rootwait rootfstype=ext4 console=tty0 fbcon=map:0 net.ifnames=0 quiet root=/dev/mmcblk0p1 rw rootwait rootfstype=ext4
12
13 # When testing a custom kernel, it is recommended that you create a backup of
14 # the original kernel and add a new entry to this file so that the device can
15 # fallback to the original kernel. To do this:
16 #
17 # 1. Make a backup of the original kernel
18 # sudo cp /boot/Image /boot/Image.backup
19 #
20 # 2. Copy your custom kernel into /boot/Image
21 #
22 # 3. Uncomment below menu setting lines for the original kernel
23 #
24 # 4. Reboot
25
26 # LABEL backup
27 # MENU LABEL backup kernel
28 # LINUX /boot/Image.backup
29 # INITRD /boot/initrd
30 # APPEND \${cbootargs}
31
32 LABEL JetsonIO
33 MENU LABEL Custom 40-pin Header Config
34 LINUX /boot/Image
35 FDT /boot/kernel_tegra210-p3448-0000-p3449-0000-b00-user-custom.dtb
36 INITRD /boot/initrd
37 APPEND tegraid=21.1.2.0.0 ddr_die=4096M@2048M section=512M memtype=0 vpr_resize usb_port_owner_info=0 lane_owner_info=0
 emc_max_dvfs=0 touch_id=0@63 video=tegrafb no_console_suspend=1 debug_uartport=lsport,4 earlyprintk=uart8250-32bit,0x70006000 maxcpus=4
 usbcore.old_scheme_first=1 lp0_vec=0x1000@0xffff780000 core_edp_mv=1125 core_edp_ma=4000 gpt earlycon=uart8250,mmio32,0x70006000 root=/dev/mmcblk0p1 rw rootwait rootfstype=ext4 console=tty0 fbcon=map:0 net.ifnames=0

Plain Text ▾ Tab Width: 8 ▾ Ln 1, Col 1 ▾ INS

In the W100 product, the 2 pwm ports on the J41 Connector will be used.

\$ sudo /opt/nvidia/jetson-io/jetson-io.py

Run the command above can configure J41 settings.

However, once the Jetson Nano save the new J41 connector configuration and reboot, the file: /boot/extlinux/extlinux.conf will be added a new LABEL and which will be selected as the default LABEL.

The J50 UART port (/dev/ttyS0) will belong to tty group again. The debug serial console will reopen. The J50 UART port CANNOT be freely used.

Ref SZ Motor Controller with 8 Documents (6/28-4/21, 2023)

CANopen standardizes communication between devices and applications from different manufacturers. It has been used in automation and motion applications

The Basics of CANopen Updated Mar 31, 2023

<https://www.ni.com/en-us/shop/seamlessly-connect-to-third-party-devices-and-supervisory-system/the-basics-of-canopen.html>



1. CAN and/or 485 interface for the motor controller

2. HL: CAN protocol over 485 hardware

From chatGPT: the CAN (Controller Area Network) protocol can be implemented on hardware that uses the RS-485 physical layer. While CAN and RS-485 are different protocols, they can be used together by incorporating a **CAN transceiver** that supports RS-485 signaling.

The RS-485 standard defines the electrical characteristics and signaling for communication over a differential pair of wires. On the other hand, CAN is a higher-level protocol that provides a robust and efficient communication method for devices on a network.

To implement CAN on RS-485 hardware, you would need a **CAN controller** and a **CAN transceiver that supports RS-485 signaling**. The CAN controller manages the CAN protocol and handles tasks such as message framing, error checking, and arbitration. The CAN transceiver ...

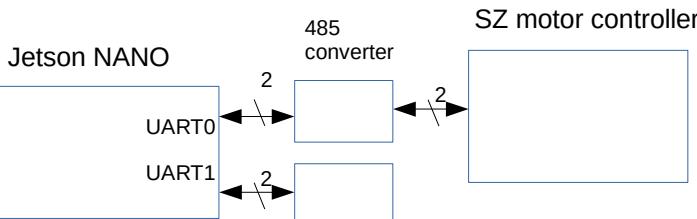
To use CAN over RS-485, you would need a transceiver that supports both protocols. This transceiver would convert the differential CAN signals to the RS-485 electrical signaling and vice versa. Additionally, you would require appropriate software or firmware to implement the CAN protocol on top of the RS-485 physical layer.

It's worth noting that while CAN and RS-485 share some similarities in their electrical signaling, they are different protocols with different features and capabilities. Adapting CAN to work over RS-485 may require additional considerations and modifications to ensure proper functionality and compatibility.

- 1How to connect to PC and use ZLAC8015(D) software.pdf
- 2ZLAC8015 CANopen Communication Routine V1.05.pdf
- 3ZLAC8015 MANUAL V2.00.pdf
- 4ZLAC8015 RS485 COMMUNICATION Version 1.03.pdf
- 5ZLAC8015_V1.0(2).eds
- 6ZLAC8015D_接线（英文）.mp4
- 7ZLTECH-HMD-MAX-V2_0.exe
- 8ZLTECH-HMD-MAX-V2_0 使用说明 .pdf

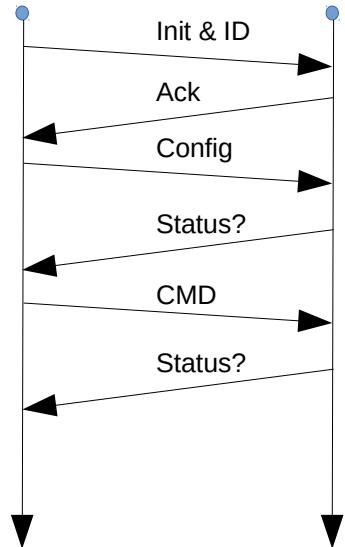
harry@workstation:/media/harry/easystore1/backup-2020-2-15/CTI/
3projects/3-8-smart-tech/3-8-4-CTI/3-8-4-6-products/AIV200/190-
robots-health/190q-manufacturing-production/190q-6-hub-motors/sz-
motor-and-controller/ZLAC8015D\$

RS485 for SZ Motor Controller (6/30,2023)



485 host
(Jetson NANO)

485 client
(motor controller)



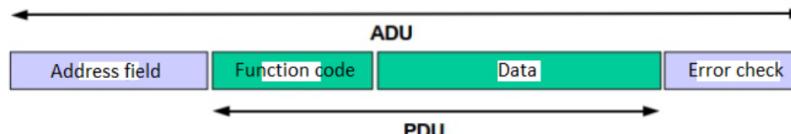
1. (a) Perform UART loopback test; (b) Use USB-485 module on Jetson NANO and laptop, connect them then send "hello" from NANO to laptop, then laptop receives the message by sending "ACK" and time stamp.

2. Run 485 motor controller code on Jetson NANO

3. Run 485 motor controller code on Jetson NANO with SZ motor controller connected for testing motor drive

putty is recommended:
<https://phoenixnap.com/kb/how-to-install-putty-on-ubuntu>

minicom for ubuntu for uart loopback test;



```
https://www.temu.com/subject/n9/googleshopping-landingpage-a-psurl.html?
goods_id=601099514148005&_bg_fs=1&_p_rfs=1&_x_ads_channel=google&
_x_ads_sub_channel=shopping&_x_login_type=Google&_x_vst_scene=advjs&sk
u_id=17592201308684&_x_ns_sku_id=17592201308684&_x_gmc_account=6
47900107&_x_ads_account=5532219654&_x_ads_set=20041590441&_x_ads
_id=14719342774&_x_ads_creative_id=656608466092&_x_ns_source=q&_x
_ns_gclid=CjOKCQjwnf-
kBhICnArisAfIg4928rZL_rKvmkK96fg2_WJjbDeJzxNfhdl5hxq65a7E_9HHloLx
HNYsaAmMeEALw_wcB&_x_ns_placement=&_x_ns_match_type=&_x_ns_ad
_position=&_x_ns_product_id=17592201308684&_x_ns_target=&_x_ns_devi
cmodel=&_x_ns_wbraid=CjgKCAjwnf-
kBhAgEigAPWRMtuJ9onOs2pqUnlal7J7ZE-
PeBbEW3_zN1Py6xyZzLOZNsQkcGgDeo&_x_ns_gbraid=0AAAAAA04mICG4
BRGVkuHTSVK_o6nYVm0&_x_ns_targetid=pla-
2078102449813&gclid=CjOKCQjwnf-
kBhICnArisAfIg4928rZL_rKvmkK96fg2_WJjbDeJzxNfhdl5hxq65a7E_9HHloLx
HNYsaAmMeEALw_wcB&_x_to_fang_tmpl_subj=ph9_ph9basesubj&adg_ctx
=f-e479ed19
```



ZLAC8015D RS485 COMMUNICATION Version 1.03.pdf



ZLAC8015D Servo Driver (Special for Hub Servo Motor) RS485 Communication Instruction Version 1.01

ZLAC8015D SERVO DRIVER (SPECIAL FOR HUB

SERVO MOTOR)

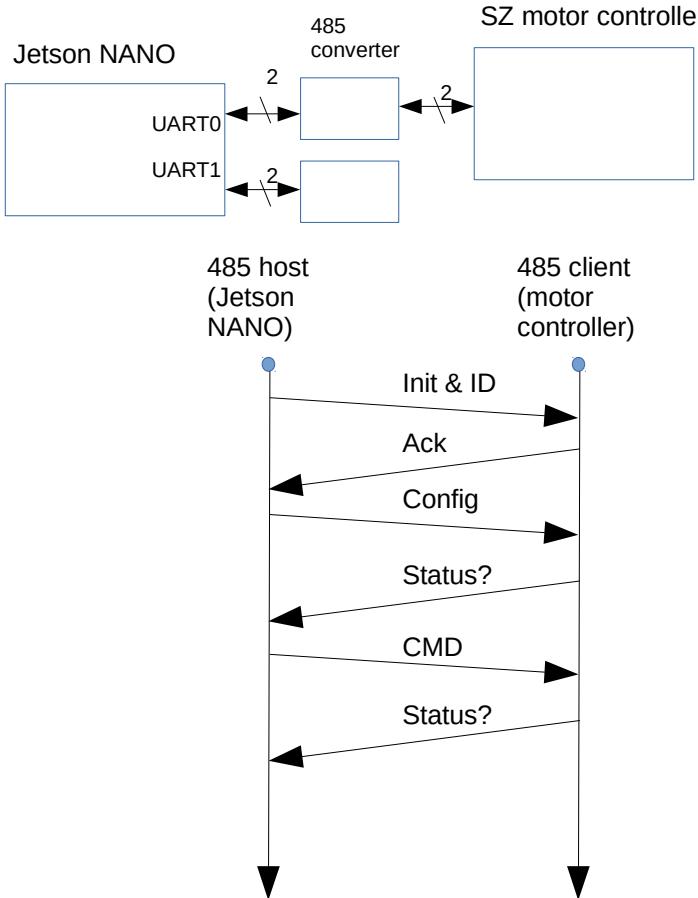
RS485 COMMUNICATION INSTRUCTION

Version	Description	Date
V1.00	First edition	
V1.01	1. Revise some control routine errors; 2. Add or delete some addresses	2021-3-23

Document:

harry@workstation:/media/harry/easystore1/backup-2020-2-15/CTI/3proejcts/3-8-smart-tech/3-8-4-CTI/3-8-4-6-products/AIV200/190-robots-health/190q-manufacturing-production/190q-6-hub-motors/sz-motor-and-controller/ZLAC8015D

Communication Protocol for the RS485 SZ Motor Controller (7/6-6/30,2023)



- Ref: 1. [/media/harry/easystore1/backup-2020-2-15/CTI/3proejcts/3-8-smart-tech/3-8-4-CTI/3-8-4-6-products/AIV200/190-robots-health/190q-manufacturing-production/190q-6-hub-motors/sz-motor-and-controller/source-9ZLAC8015D-RS485-Master-demo\\$ tree -L 2](https://media/harry/easystore1/backup-2020-2-15/CTI/3proejcts/3-8-smart-tech/3-8-4-CTI/3-8-4-6-products/AIV200/190-robots-health/190q-manufacturing-production/190q-6-hub-motors/sz-motor-and-controller/source-9ZLAC8015D-RS485-Master-demo$ tree -L 2)
- ```

.
├── 9ZLAC8015D-RS485-Master-demo.rar
│ └── CHN-version
│ └── 3_ZLAC8015D-RS485-Master-demo.rar
│ └── ZLAC8015D-RS485-Master-demo
│ ├── CORE
│ ├── HARDWARE
│ ├── keilkill.bat
│ ├── OBJ
│ ├── STM32F10x_FWLib
│ ├── SYSTEM
│ ├── USER
│ └── USMART

```
- Ref: 2. [/media/harry/easystore1/backup-2020-2-15/CTI/3proejcts/3-8-smart-tech/3-8-4-CTI/3-8-4-6-products/AIV200/190-robots-health/190q-manufacturing-production/190q-6-hub-motors/sz-motor-and-controller/ZLAC8015D\\$ tree -L 1](https://media/harry/easystore1/backup-2020-2-15/CTI/3proejcts/3-8-smart-tech/3-8-4-CTI/3-8-4-6-products/AIV200/190-robots-health/190q-manufacturing-production/190q-6-hub-motors/sz-motor-and-controller/ZLAC8015D$ tree -L 1)
- ```

.
├── 1How to connect to PC and use ZLAC8015(D) software.pdf
├── 2ZLAC8015D CANopen Communication Routine V1.05.pdf
├── 3ZLAC8015D MANUAL V2.00.pdf
└── 4ZLAC8015D RS485 COMMUNICATION Version 1.03.pdf
.
├── 5ZLAC8015D_V1.0(2).eds
├── 6ZLAC8015D 接线 (英文) .mp4
├── 7ZLTECH-HMD-MAX-V2_0.exe
├── 8ZLTECH-HMD-MAX-V2_0 使用说明 .pdf
└── 9ZLAC8015D-RS485-Master-demo.rar

```

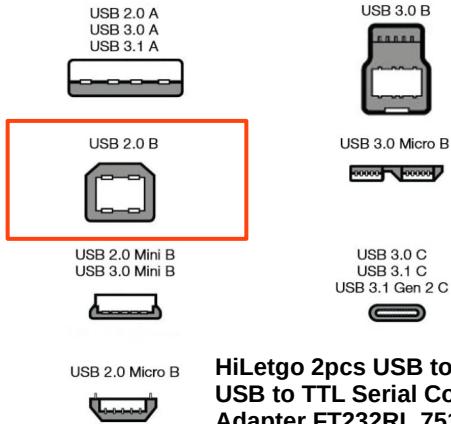
0 directories, 9 files

USB to RS485 Converter for Testing (7/6,2023)

1. USB-485 Converter with USB-B
Male connector

USB-485 Converter with USB-B
connector

USB Connection Type Reference Chart

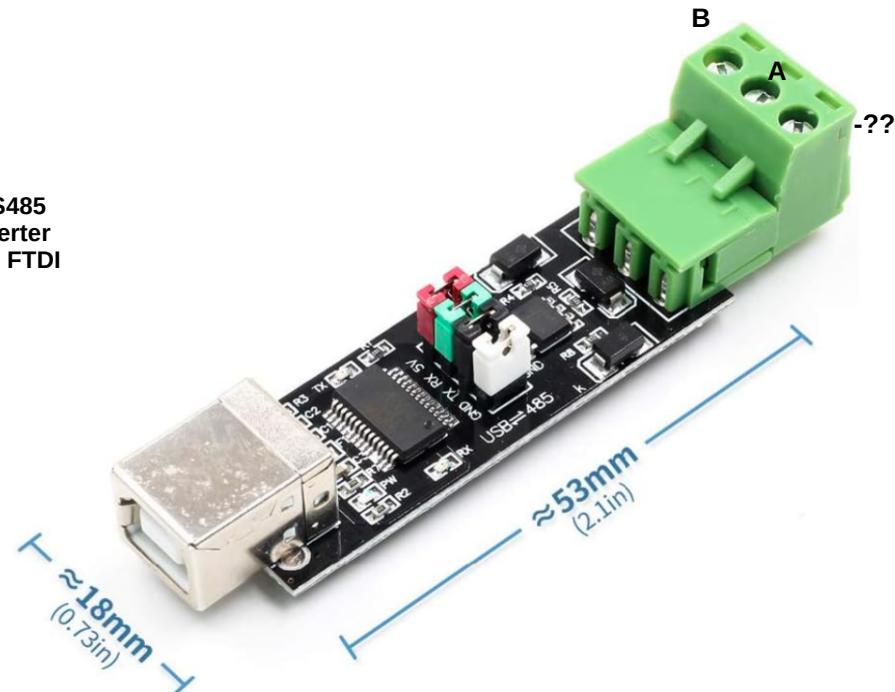


rgzhihuifz 3 Pack USB 2.0 AF/BM Adapter



30.8mm

**HiLetgo 2pcs USB to RS485
USB to TTL Serial Converter
Adapter FT232RL 75176 FTDI
Interface**



Sample Control for the RS485 SZ Motor Controller (7/6-6/30,2023)

Example:

For language translation on line: DeepL, yy

另外请提供一个用于贵公司驱动器的软件源程序样本，实现（1）初始设置；（2）转动左轮5秒，然后转动右轮5秒；（3）然后反向转动左轮5秒，再反向转动右轮5秒钟。

HL: 2023-6-30: Please also provide a sample software source program for your drive that implements (1) the initial setup, (2) turning the left wheel for 5 seconds, then turning the right wheel for 5 seconds, and (3) then reversing the left wheel for 5 seconds, then reversing the right wheel for 5 seconds.

```

01 06 20 0D 00 03 53 C8
01 06 20 0E 00 08 E2 0F
01 10 20 88 00 02 04 00 0A 00 0A C3 AD (set left motor and right motor target speed to 10RPM in positive direction)
+set 5s delay time in keil
01 10 20 88 00 02 04 FF F6 FF F6 72 38(set left motor and right motor target speed to 10RPM in negative direction)
+set 5s delay time in keil

```

01 06 20 0D 00 03 53 C8

Note: the syntax of the command (1)
address of the register (1 byte); (2) the
content going into the register (1 byte)

Drive address: 01

Flow: 1st command, select function mode
(enable) >> 2nd command, control word to
enable with E2 0F for CRC (3) write mul
byte function > 20 88 start address, 00 02
(2 regs), 04 (no. of bytes to be written) 00
0A

Function code: 06

Control mode/ velocity profile 20 0D (2 bytes addr)

??? the content: 00 03 (per table 3.1) (content)

Ref1, pp. 5

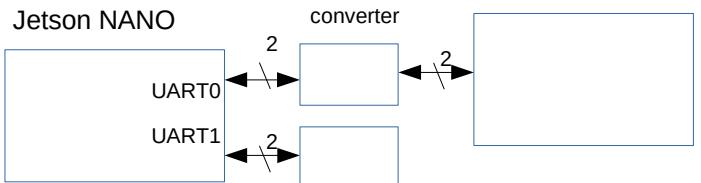
16 bit Register
(address)



High byte

Low byte

The Instruction Format for RS485 SZ Motor Controller (7/11,2023)



Example:

For language translation on line: DeepL, yy

另外请提供一个用于贵公司驱动器的软件源程序样本，实现（1）初始设置；（2）转动左轮5秒，然后转动右轮5秒；（3）然后反向转动左轮5秒，再反向转动右轮5秒钟。

HL: 2023-6-30: Please also provide a sample software source program for your drive that implements (1) the initial setup, (2) turning the left wheel for 5 seconds, then turning the right wheel for 5 seconds, and (3) then reversing the left wheel for 5 seconds, then reversing the right wheel for 5 seconds.

```
01 06 20 0D 00 03 53 C8
01 06 20 0E 00 08 E2 0F
01 10 20 88 00 02 04 00 0A 00 0A C3 AD (set left motor and right motor target speed to 10RPM in positive direction)
+set 5s delay time in keil
01 10 20 88 00 02 04 FF F6 FF F6 72 38(set left motor and right motor target speed to 10RPM in negative direction)
+set 5s delay time in keil
```

01 06 20 0D 00 03 53 C8

Note: the syntax of the command (1) address of the register (1 byte); (2) the content going into the register (1 byte)

Drive address: 01
 Function code: 06
 Control mode/ velocity profile 20 0D (2 bytes addr)
 ??? the content: 00 03 (per table 3.1) (content)

Ref1, pp. 5

Flow: 1st command, select function mode (enable) >> 2nd command, control word to enable with E2 0F for CRC (3) write multi byte function > 20 88 start address, 00 02 (2 regs), 04 (no. of bytes to be written) 00 0A

- (1) Level 1: Slave/client Address, 1 byte, the top block; (2) Level 2: Function (Command), 1 byte, the level 2 in the block diagram; (3) Level 3: Control registers address, 2 bytes; (4) Level 4: write the control pattern into the control register, 2 bytes (the control register is 16 bits, U16); (5) Level 5 (the end level): CRC code, 2 bytes.

16 bit Register (address)

High byte

Low byte

The Flow Chart for the RS485 SZ Motor Controller (7/18,2023)

Example:

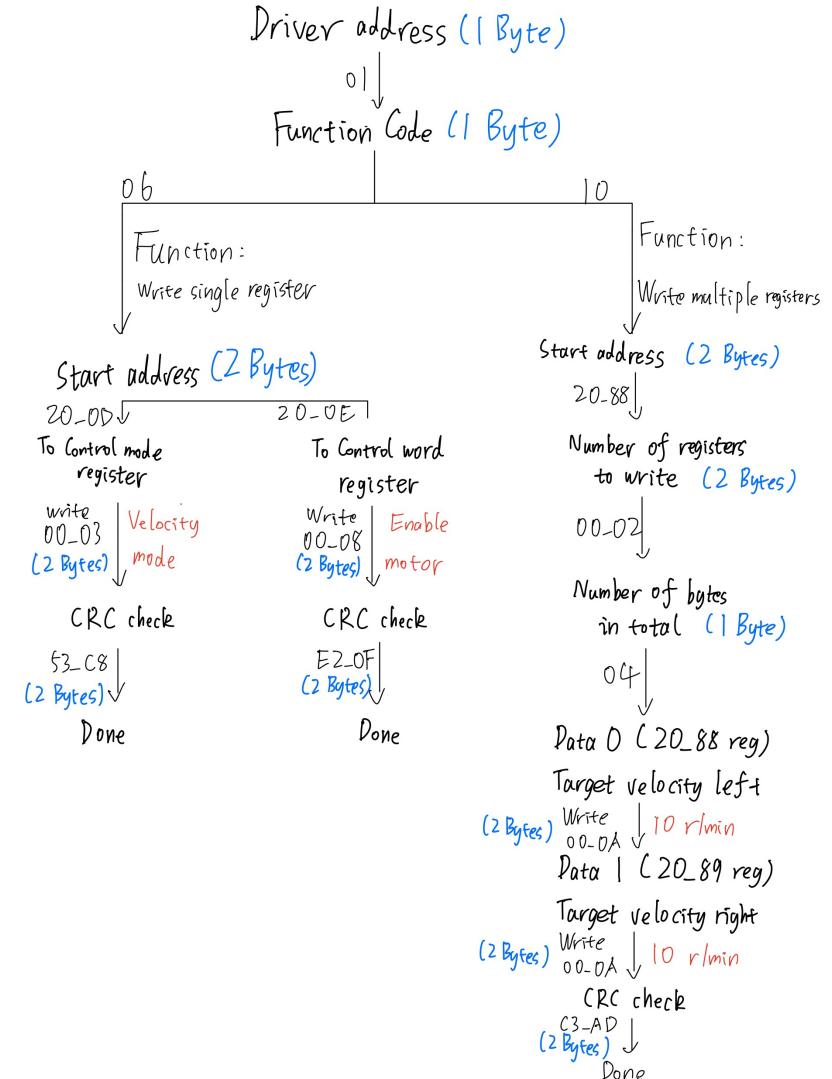
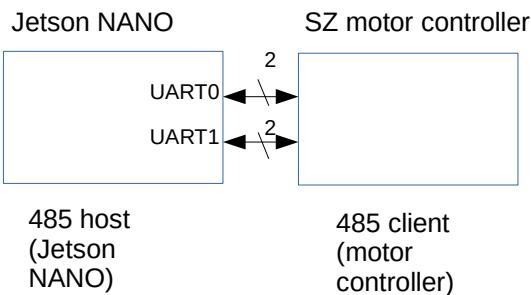
For language translation on line: DeepL, yy

另外请提供一个用于贵公司驱动器的软件源程序样本，实现（1）初始设置；（2）转动左轮 5 秒，然后转动右轮 5 秒；（3）然后反向转动左轮 5 秒，再反向转动右轮 5 秒钟。

HL: 2023-6-30: Please also provide a sample software source program for your drive that implements (1) the initial setup, (2) turning the left wheel for 5 seconds, then turning the right wheel for 5 seconds, and (3) then reversing the left wheel for 5 seconds, then reversing the right wheel for 5 seconds.

```

01 06 20 0D 00 03 53 C8
01 06 20 0E 00 08 E2 0F
01 10 20 88 00 02 04 00 0A 00 0A C3 AD (set left motor and right motor target speed to 10RPM in positive direction)
+set 5s delay time in keil
01 10 20 88 00 02 04 FF F6 FF F6 72 38(set left motor and right motor target speed to 10RPM in negative direction)
+set 5s delay time in keil
  
```





Nnn-n-Ctione-Default-SD-Card- Installations-hl-2023-3-14.odp (Under Construction)

CTI One Corporation
This document is created by:

Harry Li, Ph.D.
Yusuke Yakuwa

March 14, 2023



Ctione Default SD-Card Installations

(7/11,2023)

No.	Package Name	Purpose
-----	--------------	---------

1.		
----	--	--

..		
----	--	--

	minicom	for UART test
--	---------	---------------

Bring Up Jetson NANO (7/12,2023)

Step 1. we are using the following development kit, please check its version first. It should be B01.

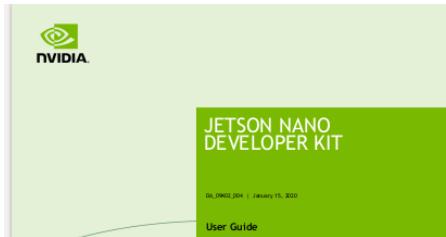
- The latest Jetson Nano Developer Kit (part number 945-13450-0000-100), which includes carrier board revision B01.
- The original Jetson Nano Developer Kit (part number 945-13450-0000-000), which includes carrier board revision A02.

Note HL 2023-7-12 Confidential:
User id: ctione
Pwd: ctione

Step 2. Insert prepared SD card with the kernel image, then connect the power, if the pwe LED light does not on, then check the jumper J48, The J48 jumper has to be placed to connect both pins to make sure the barrel jack power supply is selected.

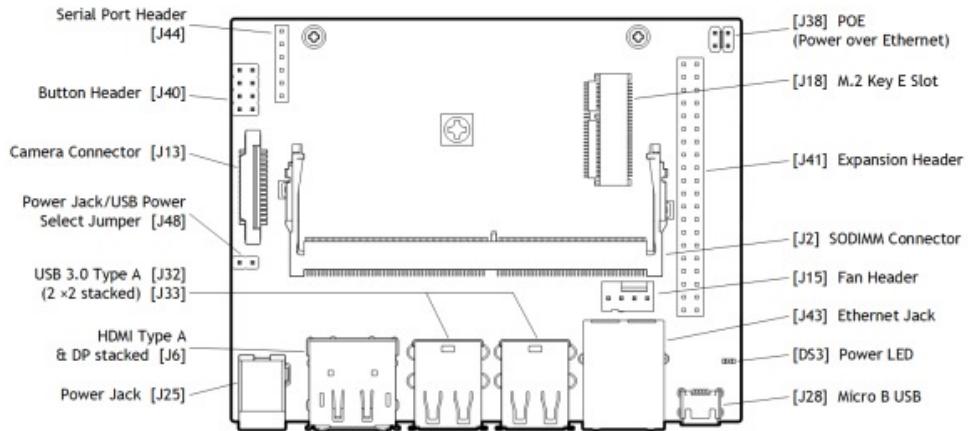
Document:

```
/media/harry/easystore1/backup-2020-2-15/CTI/3proejcts/3-8-smart-tech/3-8-4-CTI/3-8-4-6-products/AIV200/190-robots-health/190q-manufacturing-production/190q-7-integration/190q-7e-bringUp-nano/190q-7e-0-Nano-dev-user-guide$
```

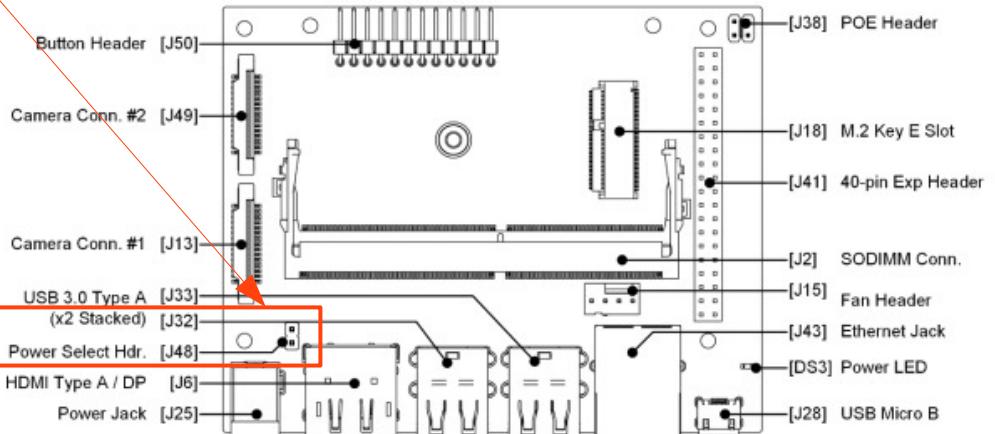


Tutorial link: <https://github.com/YahboomTechnology/Jetson-NANO-4GB>

Developer kit carrier boards: rev A02 top view



Developer kit module and carrier board: rev B01 top view



Jetson NANO UART Loopback Test (7/12,2023)

Step 1. On NANO board, find 2 UART ports, pp. 13, 190p-1b-io-board-nano-v6-hl-2023-7-09.odp
J41-8 Tx, J41-10 Rx, connect them to form a loop;

UART1(left)_1	VCC	J41-1	3V3
UART1(left)_2	TxD	J41-10	UART_2_RX
UART1(left)_3	RxD	J41-8	UART_2_TX
UART1(left)_4	GND	J41-8	gnd
UART2(left)_1	VCC	J41-1	3V3
UART2(left)_2	TxD	J50_RX	
UART2(left)_3	RxD	J50_TX	
UART2(left)_4	GND	J41-8	gnd



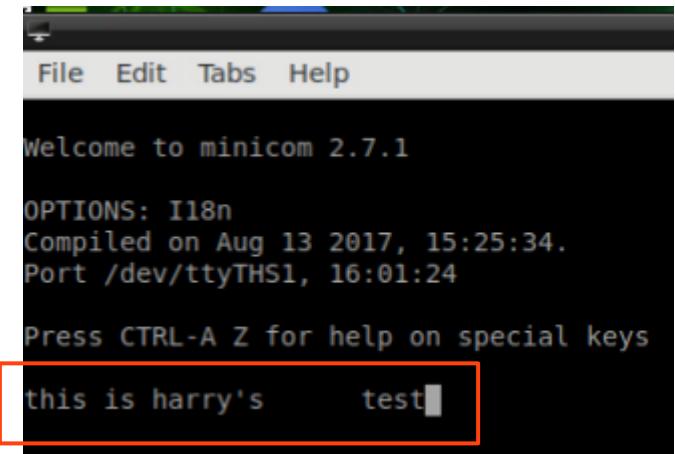
Note: change the NANO display resolution:

```
xrandr --output HDMI-0 --mode 1024x768
```

2. Find UART ports:

```
dmesg | grep tty
```

Run minicom to set up the UART communication parameters by HL.
Putty testing and Python code testing by Youran Zheng



```
File Edit Tabs Help
ctione@ctione-desktop:~$ dmesg | grep tty
[    0.000000] Kernel command line: tegraid=21.1.2.0.0 ddr_die=4096M@2048M section=512M memtype=0 vpr_
b_port_owner_info=0 lane_owner_info=0 emc max_dvfs=0 touch_id=0@63 video=tegrafb no_console_suspend=1
[    0.001841] console [tty0] enabled
[    1.128037] console [ttyS0] disabled
[    1.128145] 70006000.serial: ttyS0 at MMIO 0x70006000 (irq = 63, base_baud = 25500000) is a Tegra
[    1.128245] console [ttyS0] enabled
[    1.129216] 70006040.serial: ttyTHS1 at MMIO 0x70006040 (irq = 64, base_baud = 0) is a TEGRA_UART
[    1.129618] 70006200.serial: ttyTHS2 at MMIO 0x70006200 (irq = 65, base_baud = 0) is a TEGRA_UART
[    2.985128] systemd[1]: Created slice system-serial\x2dgetty.slice.
ctione@ctione-desktop:~$
```

Jetson NANO UART to 485 Motor Controller Test (7/12,2023)

Note: You should perform the UART loop back test first before to start this test.

Step 1. Make sure the UART to 485 converter is ready, the power Vcc = 3p3 V or 5 V.

https://www.amazon.com/HiLetgo-Reciprocal-Hardware-Automatic-Converter/dp/B082Y19KV9/ref=sr_1_4?crid=3ND4JA7TYWJEG&keywords=uart+to+485&qid=1688190894&sproduct=uart+to+485%2Caps%2C136&sr=8-4

SZ Motor Controller Wiring (Source: 6ZLAC8015D 接线 (英文) .mp4)

Orange	A
Brown	B

Field Service Engineer: Laptop Based RS232 Loopback Test (7/12,2023)

Step 1. On Laptop, UART loopback test.

Hardware: need UART-RS232 converter

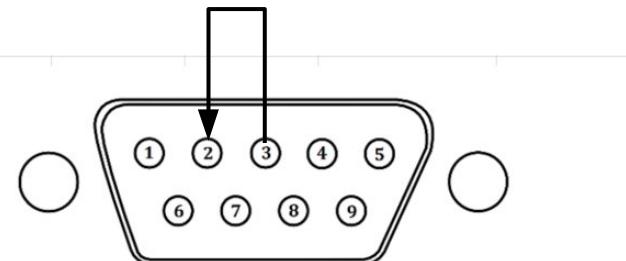
Software:

1. Find USB-to-UART (ttyx) devices:

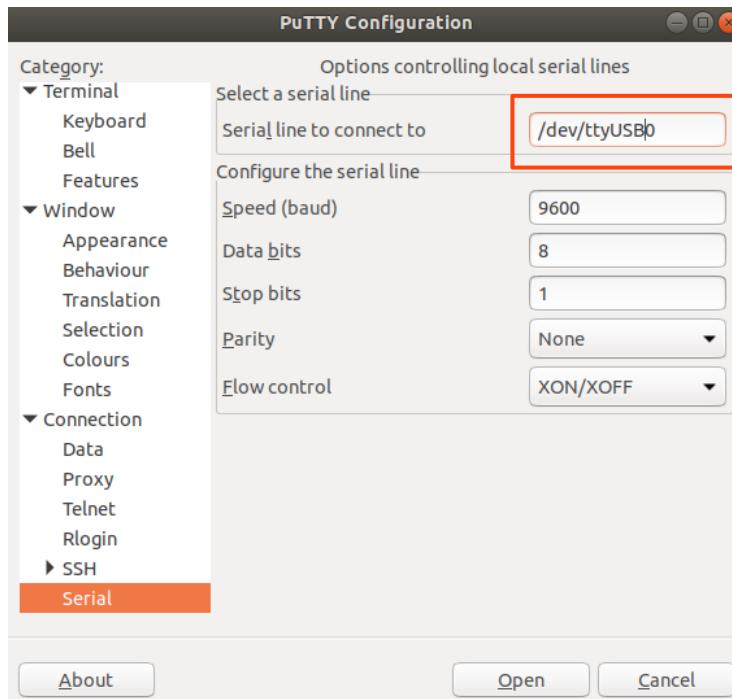
```
dmesg | grep tty
```

```
File Edit View Search Terminal Help
harry@workstation: ~
(base) harry@workstation:~$ dmesg | grep tty
[ 0.101516] printk: console [tty0] enabled
[ 4768.370139] usb 1-10.1.3: FTDI USB Serial Device converter now attached to /dev/ttyUSB0
```

2. Connect pin 2 (Rx) and 3 (Tx), now start putty, see Fig. 2.



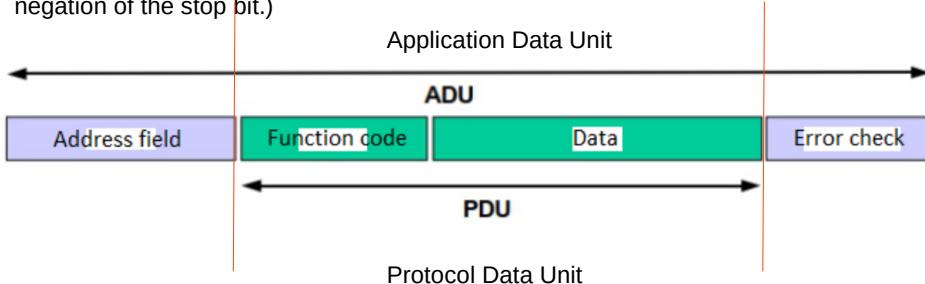
	RS232	CAN	CAN (OBD-II)	RS485
PIN1			GND	A
PIN 2	RX	CANL		B
PIN 3	TX	GND	CANH	
PIN 5	GND		CANL	GND
PIN 7		CANH		



Modbus RTU for the RS485 Motor Controller (7/6-6/30,2023)

HL 2023-7-10: from SZ manufacturer's document. The conditions for the controller setting:

1. Modbus RTU protocol.
2. The **driver address can be set to 0-127, the default address is 1.**
3. Baud rate: 115200 by default (ZLAC8015D has 7 optional baud rates: 9600, 19200, 38400, 57600, 115200, 128000, 256000).
4. There are 8 data bits, **No Parity**, Stop bit is 1, (HL: the start bit is the negation of the stop bit.)



Ref1.



ZLAC8015D Servo Driver [Special for Hub Servo Motor] RS485 Communication Instruction Version 1.03

ZLAC8015D SERVO DRIVER (SPECIAL FOR HUB

SERVO MOTOR)

RS485 COMMUNICATION INSTRUCTION

1 <https://www.youtube.com/watch?v=6MtFLA0NW1c>

default Modbus port 502 in raspberry pi

2 <https://pymodbus.readthedocs.io/en/latest/> Requires Python >= 3.8

Pymodbus installation and sample code

Simple synchronous client

```

#!/usr/bin/env python
"""
PyModbus synchronous client example.

An example of a single threaded synchronous client.

usage: simple_client_async.py

All options must be adapted in the code
the corresponding server must be started before e.g. as:
python3 server_sync.py

# -----
# import the various client implementations
# -----
from pymodbus import pymodbus_synchronous
from pymodbus.client import (
    ModbusSerialClient,
    ModbusTcpClient,
    ModbusTlsClient,
    ModbusUdpClient,
)
from pymodbus.exceptions import ModbusException
from pymodbus.pdu import ExceptionResponse
from pymodbus.transaction import (
    ModbusAsciiFramer,
    ModbusBinaryFramer,
    ModbusRtuFramer,
    ModbusSocketFramer,
    ModbusTlsFramer,
)

def run_sync_client(host, port):
    """
    Run sync client.
    """

    # activate debugging
    pymodbus_apply_logging_config("DEBUG")

    # change to test other client types

```

Ref2.

<https://buildmedia.readthedocs.org/media/pdf/pymodbus/latest/pymodbus.pdf>

PyModbus
Release 3.4.0dev

Open Source volunteers

Ref3.

https://github.com/ahanjaya/Modbus-RTU/blob/main/modbus_min.py



Nnn-n-product-liability-insurance-hl-2023-7-17.odp

(Under Construction)

CTI One Corporation
This document is created by:

Harry Li, Ph.D.
Yusuke Yakuwa

July 17, 2023

Product Liability Insurance (7/17, 2023)

www.wixsite.com
Login Account: contact@ctione.com
Password: Biophotonics06!

<https://www.insurancecanopy.com/product-liability-medical-devices>

 Insurance Canopy
<https://www.insurancecanopy.com> › product-liability-... ::

Product Liability Insurance for Medical Devices

Typically, policy premiums start around \$2,298 per year for medical equipment insurance. premium can vary based on different factors, such as: Business ...

Product Liability Insurance Cost

May 18, 2023 – Product liability insurance is typically included in general liability insurance, which costs small businesses an average of \$42 per month.

[Product Liability Insurance - Buy Instantly From \\$19 a Month](#)

HDMI Bike Mount Monitor (7/17,2023)

**10.1 or 7 Inch Small Monitor, Raspberry Pi Monitor
Built-in Speakers, Mini Monitor 1024 x 600, HDMI
Monitor 60 Hz, 5ms, IPS Screen Display w/Remote
Control, HDMI VGA AV BNC Input, Computer
Monito \$49 for 7", 52.99 for 10.1"**

www.wixsite.com
Login Account: contact@ctione.com
Password: Biophotonics06!

https://www.amazon.com/Monitor-Raspberry-Speakers-Display-Computer/dp/B09ZDMNZPT/ref=asc_df_B09ZDMNZPT/?tag=hyprod-20&linkCode=df0&hvadid=615861270911&hvpos=&hvnetw=g&hvrand=1458690738210876213&hvpone=&hvptwo=&hvqmt=&hvdev=c&hvdvcmdl=&hvlocint=&hvlo cphy=9031968&hvtargid=pla-1835315613251&psc=1



NVIDIA Jetson Nano J41 Header Pinout

<https://www.jetsonhacks.com/nvidia-jetson-nano-j41-header-pinout/>

I2C and UART pins are connected to hardware and should not be reassigned. By default, all other pins (except power) are assigned as GPIO. Pins labeled with other functions are recommended functions if using a different device tree.

	GND	25	26	SPI_1_CS1	gpio20
	I2C_1_SDA I2C Bus 0	27	28	I2C_1_SCL I2C Bus 0	
gpio149	CAM_AF_EN	29	30	GND	
gpio200	GPIO_PZ0	31	32	LCD_BL_PWM	gpio168
gpio38	GPIO_PE6	33	34	GND	
gpio76	I2S_4_LRCK	35	36	UART_2_CTS	gpio51
gpio12	SPI_2_MOSI	37	38	I2S_4_SDIN	gpio77
	GND	39	40	I2S_4_SDOUT	gpio78

gpio-nano-2021-10-14

Sysfs GPIO	Name	Pin	Pin	Name	Sysfs GPIO
	3.3 VDC Power	1	2	5.0 VDC Power	
	I2C_2_SDA I2C Bus 1	3	4	5.0 VDC Power	
	I2C_2_SCL I2C Bus 1	5	6	GND	
gpio216	AUDIO_MCLK	7	8	UART_2_TX <code>/dev/ttyTHS1</code>	
	GND	9	10	UART_2_RX <code>/dev/ttyTHS1</code>	
gpio50	UART_2_RTS	11	12	I2S_4_SCLK	gpio79
gpio14	SPI_2_SCK	13	14	GND	
gpio194	LCD_TE	15	16	SPI_2_CS1	gpio232
	3.3 VDC Power	17	18	SPI_2_CS0	gpio15
gpio16	SPI_1_MOSI	19	20	GND	
gpio17	SPI_1_MISO	21	22	SPI_2_MISO	gpio13
gpio18	SPI_1_SCK	23	24	SPI_1_CS0	gpio19
	GND	25	26	SPI_1_CS1	gpio20