AIO Library

2014-06-21

Content

1 Install library

1.1 X86:

1.1.1 copy x86_gcc.tar to PC

1.1.2 tar –xvf x86_gcc.tar

1.1.3 libaio_x86.so will be extracted to /usr/local/lib

1.2 Matrix tool chain arm-linux-gnueabi-gcc: (AT9G20 and AT9G45 SoC)

1.2.1 copy arm_eabigcc.tar to PC

1.2.2 tar –xvf arm_eabigcc.tar

1.2.3 libaio_9G20.so will be extracted to /usr/local/arm-linux-gnueabi/lib/

1.3 Matrix tool chain arm-linux- gcc: (AT9200 SoC)

1.3.1 Include the file libaio_9200.so to your software program

2 Make example

2.1 X86:

2.1.1 make –f makefile_gcc

2.1.2 the execution file will be at ./output/gcc

2.2 Matrix arm-linux-gnueabi-gcc:

2.2.1 make –f makefile_eabigcc

2.2.2 The execution file will be at ./output/eabigcc

2.3 Matrix arm-linux- gcc:

2.3.1 make –f makefile_armgcc

2.3.2 the execution file will be at ./output/armgcc

2.3.3 libaio_9200.so is placed at ./output/armgcc/armgcc_lib

3 API Reference

3.1 IO

3.1.1 AIO_Init

3.1.1.1 int AIO_Init(void)

4 Return Value:

Succeed	AIO_OK
Fail	Refer to AIO.h

4.1.1 AIO_Connect

4.1.1.1 int AIO_Connect(char *ip_str, int port, int tmout_ms, int *socket_fd)

Arguments:

char *ip_str [in]	Device IP address
int port [in]	Modbus TCP listen port
int tmout_ms [in]	connect response timeout. The unit is in milliseconds.
int *socket_fd [out]	Socket file handle of the connection

Return Value:

Succeed	AIO_OK
Fail	Refer to AIO.h

4.1.2 AIO_Close

4.1.2.1 int AIO_Close(int socket_fd)

Arguments:

int socket_fd [in]	file handle for modbus connection

Return Value:

Succeed	AIO_OK
Fail	Refer to AIO.h

4.1.3 AIO_RIO_2018_DO_Read

4.1.3.1 int AIO_RIO_2018_DO_Read(int socket_fd, int tmout_ms, unsigned char *value)

Arguments:

int socket_fd [in]	file handle for modbus connection
int tmout_ms [in]	read response timeout. The unit is in milliseconds.
unsigned char *value [out]	stores the DO value only one DO for RIO-2018

Return Value:

Succeed	AIO_OK
Fail	Refer to RIO_2018_AIO.h

4.1.4 AIO_RIO_2018_DO_Write

4.1.4.1 int AIO_RIO_2018_DO_Write(int socket_fd, int tmout_ms, unsigned char value)

Arguments:

int socket_fd [in]	file handle for modbus connection
int tmout_ms [in]	read response timeout. The unit is in

	milliseconds.
unsigned char value [in]	stores the DO value

Return Value:

Succeed	AIO_OK
Fail	Refer to RIO_2018_AIO.h

4.1.5 AIO_RIO_2018_DI_Read

4.1.5.1 int AIO_RIO_2018_DI_Read(int socket_fd, int tmout_ms, unsigned char start_no, unsigned char quantity, unsigned char *value_buf, int buf_size, int * value_byte_count)

4.1.5.2 Response data example

4.1.5.2.1 value_buf[0]:

bit0- the first DI, bit1- the second DI

Arguments:

int socket_fd [in]	file handle for modbus connection
int tmout_ms [in]	read response timeout. The unit is in milliseconds.
int start_no [in]	the number of the first DI. DI#1: 1
int quantity [in]	quantity of DI(The maximum number is 2)
unsigned char * value_buf [out]	an array that stores the DI value.
int buf_size [in]	size of the value _buf
int * value_byte_count [out]	the response data length

Return Value:

Succeed	AIO_OK
Fail	Refer to RIO_2018_AIO.h

4.1.6 AIO_RIO_2018_TC_Read

4.1.6.1 int AIO_RIO_2018_TC_Read(int socket_fd, int tmout_ms, unsigned char start_channel, unsigned char quantity,unsigned char *value_buf, int buf_size, int * value_byte_count)

4.1.6.2 AI response see “RIO Modbus Function List” and aio example

Arguments:

int socket_fd [in]	file handle for modbus connection
int tmout_ms [in]	read response timeout. The unit is in milliseconds.
int start_ channel [in]	the channel of the first TC， TC#1: 1

int quantity [in]	quantity of TC(The maximum number is 3)
unsigned char *value_buf [out]	an array that stores the TC value
int buf_size [in]	size of the response_buf
int * value_byte_count [out]	the response data length

Return Value:

Succeed	AIO_OK
Fail	Refer to RIO_2018_AIO.h

4.1.7 AIO_RIO_2017_DO_Read

4.1.7.1 int AIO_RIO_2017_DO_Read(int socket_fd, int tmout_ms, unsigned char *value)

Arguments:

int socket_fd [in]	file handle for modbus connection
int tmout_ms [in]	read response timeout. The unit is in milliseconds.
unsigned char *value [out]	stores the DO value only one DO for RIO-2017

Return Value:

Succeed	AIO_OK
Fail	Refer to RIO_2017_AIO.h

4.1.8 AIO_RIO_2017_DO_Write

4.1.8.1 int AIO_RIO_2017_DO_Write(int socket_fd, int tmout_ms, unsigned char value)

Arguments:

int socket_fd [in]	file handle for modbus connection
int tmout_ms [in]	read response timeout. The unit is in milliseconds.
unsigned char value [in]	stores the DO value

Return Value:

Succeed	AIO_OK
Fail	Refer to RIO_2017_AIO.h

4.1.9 AIO_RIO_2017_AI_Read

4.1.9.1 int AIO_RIO_2017_AI_Read(int socket_fd, int tmout_ms, unsigned char start_channel, unsigned char quantity,unsigned

char *value_buf, int buf_size, int * value_byte_count)

4.1.9.2 AI response see “RIO Modbus Function List” and aio example

Arguments:

int socket_fd [in]	file handle for modbus connection
int tmout_ms [in]	read response timeout. The unit is in milliseconds.
int start_ channel [in]	the channel of the first AI， AI#1: 1
int quantity [in]	quantity of AI(The maximum number is 8)
unsigned char *value_buf [out]	an array that stores the AI value
int buf_size [in]	size of the response_buf
int * value_byte_count [out]	the response data length

Return Value:

Succeed	AIO_OK
Fail	Refer to RIO_2017_AIO.h

4.1.10 AIO_RIO_2010_DO_Read

4.1.10.1 int AIO_RIO_2010_DO_Read(int socket_fd, int tmout_ms, unsigned char start_no, unsigned char quantity, unsigned char * value)

Arguments:

int socket_fd [in]	file handle for modbus connection
int tmout_ms [in]	read response timeout. The unit is in milliseconds.
int start_no [in]	the number of the first DO. DO#1: 1
int quantity [in]	quantity of DO(The maximum number is 8)
unsigned char *value [out]	stores the DO value bit0-first DO…

Return Value:

Succeed	AIO_OK
Fail	Refer to RIO_2010_AIO.h

4.1.11 AIO_RIO_2010_DO_Write

4.1.11.1 int AIO_RIO_2010_DO_Write(int socket_fd, int tmout_ms, unsigned char start_no, unsigned char quantity, unsigned char value)

Arguments:

int socket_fd [in]	file handle for modbus connection

int tmout_ms [in]	read response timeout. The unit is in milliseconds.
int start_no [in]	the number of the first DO. DO#1: 1
int quantity [in]	quantity of DO(The maximum number is 8)
unsigned char value [in]	stores the DO value bit0-first DO…

Return Value:

Succeed	AIO_OK
Fail	Refer to RIO_2010_AIO.h

4.1.12 AIO_RIO_2010_DO_Write_bit

4.1.12.1 int AIO_RIO_2010_DO_Write_bit(int socket_fd, int tmout_ms, unsigned char start_no, unsigned char value)

Arguments:

int socket_fd [in]	file handle for modbus connection
int tmout_ms [in]	read response timeout. The unit is in milliseconds.
int start_no [in]	the number of the first DO. DO#1: 1
unsigned char value [in]	bit0: stores the DO value

Return Value:

Succeed	AIO_OK
Fail	Refer to RIO_2010_AIO.h

4.1.13 AIO_RIO_2010_DI_Read

4.1.13.1 int AIO_RIO_2010_DI_Read(int socket_fd, int tmout_ms, unsigned char start_no, unsigned char quantity, unsigned char *value_buf, int buf_size, int * value_byte_count)

4.1.13.2 Response data example

4.1.13.2.1 value_buf[0]:

bit0- the first DI, bit7- the eighth DI

4.1.13.2.2 value_buf[1]:

bit0- the 9th DI, bit7-the 16th DI

Arguments:

int socket_fd [in]	file handle for modbus connection
int tmout_ms [in]	read response timeout. The unit is in milliseconds.

int start_no [in]	the number of the first DI. DI#1: 1
int quantity [in]	quantity of DI(The maximum number is 16)
unsigned char * value_buf [out]	an array that stores the DI value.
int buf_size [in]	size of the value _buf
int * value_byte_count [out]	the response data length

Return Value:

Succeed	AIO_OK
Fail	Refer to RIO_2010_AIO.h

4.1.14 AIO_RIO_2010_Sensor_Read

4.1.14.1 int AIO_RIO_2010_Sensor_Read(int socket_fd, int tmout_ms, unsigned char *value_buf, int buf_size, int *response_byte_count)

4.1.14.2 AI response see “RIO Modbus Function List” and aio example

Arguments:

int socket_fd [in]	file handle for modbus connection
int tmout_ms [in]	read response timeout. The unit is in milliseconds.
unsigned char *value_buf [out]	an array that stores the sensor value
int buf_size [in]	size of the response_buf
int *response_byte_count [out]	the response data length

Return Value:

Succeed	AIO_OK
Fail	Refer to RIO_2010_AIO.h

4.2 Configure

4.2.1 AIO_Cfg_Login

4.2.1.1 int AIO_Cfg_Login(char *ip_str, int port, int tmout_ms, char *password, int *socket_fd)

Arguments:

char *ip_str [in]	Device IP address

int port [in]	Command mode TCP listen port
int tmout_ms [in]	connect response timeout. The unit is in milliseconds.
char *password	Command mode password.
int *socket_fd [out]	Socket file handle of the connection

Return Value:

Succeed	AIO_OK
Fail	Refer to RIO_2017_AIO.h

4.2.2 AIO_Cfg_RIO_2018_Get_TC_Enable

4.2.2.1 int AIO_Cfg_RIO_2018_Get_TC_Enable(int socket_fd, int tmout_ms, int start_channel, int quantity, unsigned char *response_buf, int buf_size, int *response_byte_count)

4.2.2.2 Response data example

4.2.2.2.1 response_buf[0]:

bit0- the first TC, bit2- the third TC

Arguments:

int socket_fd [in]	file handle for command mode connection
int tmout_ms [in]	read response timeout. The unit is in milliseconds.
int start_channel [in]	the channel of the first TC. TC#1: 1
int quantity [in]	quantity of TC
unsigned char *response_buf [out]	an array that stores the TC enable value or error response
int buf_size [in]	size of the response_buf
int *response_byte_count [out]	the response data length

Return Value:

Succeed	AIO_OK
Fail	Refer to RIO_2018_AIO.h

4.2.3 AIO_Cfg_RIO_2018_Set_TC_Enable

4.2.3.1 int AIO_Cfg_RIO_2018_Set_TC_Enable(int socket_fd, int tmout_ms, int start_channel, int quantity, int set_value)

4.2.3.2 set value example

4.2.3.2.1 bit0- the first TC, bit7- the third TC

Arguments:

int socket_fd [in]	file handle for command mode connection
int tmout_ms [in]	read response timeout. The unit is in milliseconds.
int start_channel [in]	the channel of the first TC. TC#1: 1
int quantity [in]	quantity of TC
int set_value	TC enable value

Return Value:

Succeed	AIO_OK
Fail	Refer to RIO_2018_AIO.h

4.2.4 AIO_Cfg_RIO_2018_Get_TC_Unit

4.2.4.1 int AIO_Cfg_RIO_2018_Get_TC_ Unit (int socket_fd, int tmout_ms, unsigned char *response_buf, int buf_size, int *response_byte_count)

4.2.4.2 Response data example

4.2.4.2.1 response_buf[0]:

0x00: degree Celsius, 0x01: degree Fahrenheit

Arguments:

int socket_fd [in]	file handle for command mode connection
int tmout_ms [in]	read response timeout. The unit is in milliseconds.
unsigned char *response_buf [out]	an array that stores the TC unit value or error response
int buf_size [in]	size of the response_buf
int *response_byte_count [out]	the response data length

Return Value:

Succeed	AIO_OK
Fail	Refer to RIO_2018_AIO.h

4.2.5 AIO_Cfg_RIO_2018_Set_TC_Unit

4.2.5.1 int AIO_Cfg_RIO_2018_Set_TC_Unit (int socket_fd, int tmout_ms, int set_value)

4.2.5.2 set value example

4.2.5.2.1 0x00: degree Celsius, 0x01: degree Fahrenheit

Arguments:

int socket_fd [in]	file handle for command mode connection

int tmout_ms [in]	read response timeout. The unit is in milliseconds.
int set_value	TC Unit value

Return Value:

Succeed	AIO_OK
Fail	Refer to RIO_2018_AIO.h

4.2.6 AIO_Cfg_RIO_2017_Get_AI_Enable

4.2.6.1 int AIO_Cfg_RIO_2017_Get_AI_Enable(int socket_fd, int tmout_ms, int start_channel, int quantity, unsigned char *response_buf, int buf_size, int *response_byte_count)

4.2.6.2 Response data example

4.2.6.2.1 response_buf[0]:

bit0- the first AI, bit7- the eighth AI

Arguments:

int socket_fd [in]	file handle for command mode connection
int tmout_ms [in]	read response timeout. The unit is in milliseconds.
int start_channel [in]	the channel of the first AI. AI#1: 1
int quantity [in]	quantity of AI
unsigned char *response_buf [out]	an array that stores the AI enable value or error response
int buf_size [in]	size of the response_buf
int *response_byte_count [out]	the response data length

Return Value:

Succeed	AIO_OK
Fail	Refer to RIO_2017_AIO.h

4.2.7 AIO_Cfg_RIO_2017_Set_AI_Enable

4.2.7.1 int AIO_Cfg_RIO_2017_Set_AI_Enable(int socket_fd, int tmout_ms, int start_channel, int quantity, int set_value)

4.2.7.2 set value example

4.2.7.2.1 bit0- the first AI, bit7- the eighth AI

Arguments:

int socket_fd [in]	file handle for command mode connection
int tmout_ms [in]	read response timeout. The unit is in

	milliseconds.
int start_channel [in]	the channel of the first AI. AI#1: 1
int quantity [in]	quantity of AI
int set_value	AI enable value

Return Value:

Succeed	AIO_OK
Fail	Refer to RIO_2017_AIO.h

4.2.8 AIO_Cfg_RIO_2017_Get_AI_Range

4.2.8.1 int AIO_Cfg_RIO_2017_Get_AI_Range(int socket_fd, int tmout_ms, int start_channel, int quantity, unsigned char *response_buf, int buf_size, int *response_byte_count)

4.2.8.2 Response data example

4.2.8.2.1 response_buf[0]: the first AI

4.2.8.2.2 response_buf[1]: the second AI

Arguments:

int socket_fd [in]	file handle for command mode connection
int tmout_ms [in]	read response timeout. The unit is in milliseconds.
int start_ channel [in]	the channel of the first AI. AI#1: 1
int quantity [in]	quantity of AI
unsigned char *response_buf [out]	an array that stores the AI range value or error response
int buf_size [in]	size of the response_buf
int *response_byte_count [out]	the response data length

Return Value:

Succeed	AIO_OK
Fail	Refer to RIO_2017_AIO.h

4.2.9 AIO_Cfg_RIO_2017_Set_AI_Range

4.2.9.1 int AIO_Cfg_RIO_2017_Set_AI_Range(int socket_fd, int tmout_ms, int start_channel, int quantity, unsigned char *request_buf, int buf_size)

4.2.9.2 request data example

4.2.9.2.1 request_buf[0]: the first AI

4.2.9.2.2 request_buf[1]: the second AI

Arguments:

int socket_fd [in]	file handle for command mode connection
int tmout_ms [in]	read response timeout. The unit is in milliseconds.
int start_channel [in]	the channel of the first AI. AI#1: 1
int quantity [in]	quantity of AI
unsigned char *request_buf [out]	an array that stores the AI range value or error request
int buf_size [in]	size of the request_buf

Return Value:

Succeed	AIO_OK
Fail	Refer to RIO_2017_AIO.h

4.2.10 AIO_Cfg_SaveExit

4.2.10.1.1 int AIO_Cfg_SaveExit(int socket_fd):

4.2.10.1.1.1 save the changes and reboot

Arguments:

int socket_fd [in]	file handle for command mode connection

Return Value:

Succeed	AIO_OK
Fail	Refer to RIO_2017_AIO.h

4.2.11 AIO_Cfg_Exit

4.2.11.1 int AIO_Cfg_Exit(int socket_fd)

Arguments:

int socket_fd [in]	file handle for command mode connection

Return Value:

Succeed	AIO_OK
Fail	Refer to RIO_2017_AIO.h