# AVA RF OVERVIEW

## AVA Actuators RF Series

The AVA smart actuator series can be supplied, as shown, with an RF antenna mounted for applications where the actuator is in a hard to reach area or is in areas where there is local power but no means to connect control and or feedback cables. The AVA RF series allows the user to control the opening and closing of actuators from up to and in excess of 1.5Km away. The impressive range is achieved using not only our Smart actuator but smart technology via either OCEAN or LORA frequency.

The actuators can be controlled on a one to one connection or one controller can control multiple actuators at once. The controlled can be a portable fob or more robust control box, designed by our AVA technicians and engineers. The range is available from 20Nm to 400Nm. This document acts as an overview for the range including options and setup within the onboard firmware. Available frequencies include RF433/869/915Mhz and FSK, LORA and RF OCEAN (single control). **Max distance is 3Km.**

COMING SOON

Please note that soon we will have a new app design that will allow our customers to access the smart menu that you can currently access via firmware on the actuator itself including number of cycles and any errors recorded. We hope to have this launched later this year, the Bluetooth connectivity will also allow the user to make changes to multiple actuators at once.

## "What is LoRa?"

Whenever someone asks us this question, it's hard to know exactly how to answer without knowing *why* they're asking. That's because LoRa can refer to more than one thing:

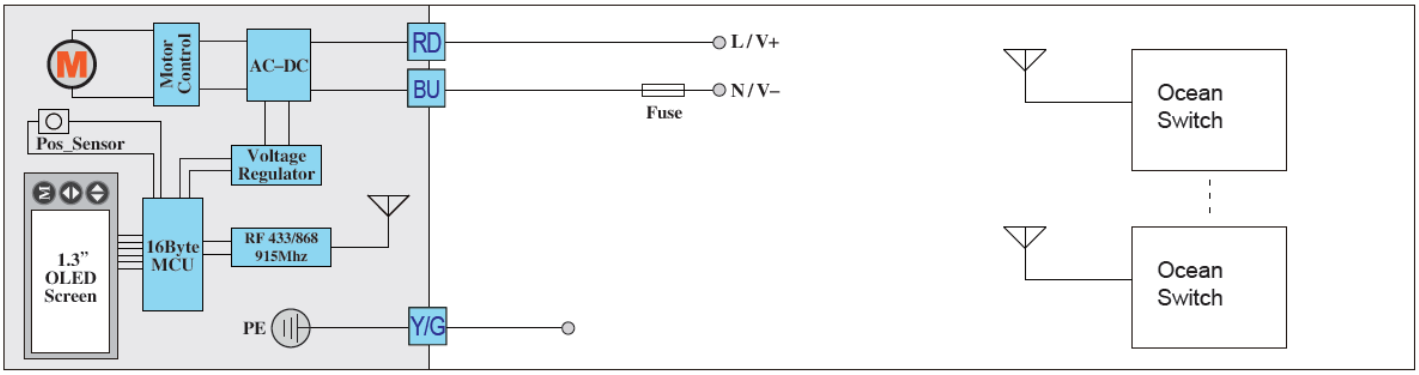
Technically, it is a radio modulation scheme—a way of manipulating a radio wave to encode information using a chirped, multi-symbol format. L*oRa* also refers to the systems that support the modulation, including LoRa chips and gateways. Sometimes it refers to the LoRa communication network for IoT applications.

LoRa, essentially, is a clever way to get very good receiver sensitivity and low bit error rate (BER) from inexpensive chips. That means low-data rate applications can get much longer range using LoRa rather than using other comparably priced radio technologies.

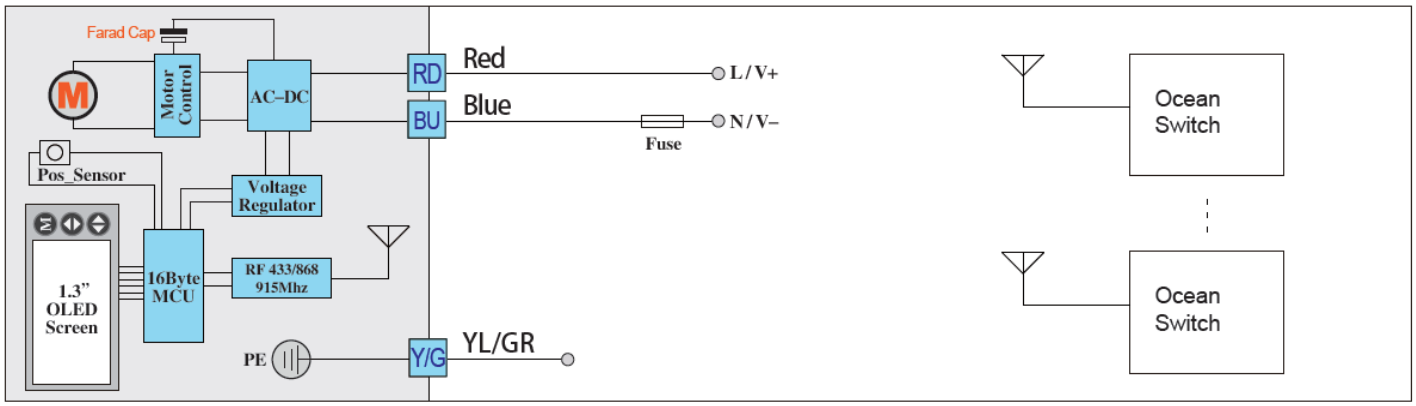
You can find more information here: https://www.link-labs.com/blog/what-is-lora

# AVA RF OVERVIEW

## ON OFF WIRING

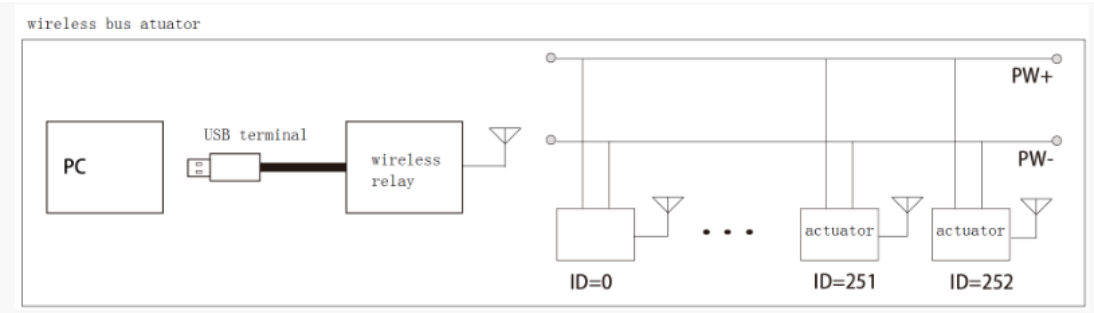



## FAILSAFE VIA CAPACITOR WIRING





**STANDARD PIN CONNECTOR SUPPLIED WITH ACTUATORS - NOTE THAT THIS CAN BE REMOVED AND CABLES TERMINATED, IT DOES *NOT* INVALIDATE YOUR WARRANTY.**




Example of Modbus RF system with multiple actuators connected.

Example of RF fob used to control opening and closing of actuator.

## RTU Mode：Host demand data message formats：

Lead code:ADDR FUNC LEN D0...Dn-1 CRC

[ADDR]:Slaver address

[FUNC]:Function code： [XX]

[LEN]: Length of [D0…Dn-1] [YY]

[D0…Dn-1]: Information content LEN x 8bit [DAT0…DATn-1]

[CRC]: CRC16 code,host calculation leads to CRC code [CRCL CRCH]

Notice：The function formats data listed in this document are all hexadecimal.

Lead code：FC FC FC F9，lead code doesn't participate in CRC operation

Open lead code could increase communication anti-interference accuracy.

| Register Addr (hexadecimal) | Name | Read write state | Data length | Data content | Detailed account |
|---|---|---|---|---|---|
| 00 01 | Actuator Status | R | 1 | specific data as follows: | ### |
| | | | | 0x00 AA | actuator acting command_ON |
| | | | | 0x00 BB | actuator acting command_OFF |
| | | | | 0x00 AC | actuator acted ON |
| | | | | 0x00 BC | actuator acted OFF |
| | | | | 0x00 BA | actuator acting command_BRAKE |
| | | | | 0x00 EA | failed alert |
| 00 03 | Valve circulation mode | R | 1 | specific data as follows: | |
| | | | | 0x00 22 | |
| | | | | | |
| 00 04 | Valve control command | R/W | 1 | specific data as follows: | |
| | | | | 0x00 01 | Current Comand Valve-On |
| | | | | 0x00 07 | Current Comand Valve-Off |
| | | | 1 | 0x00 BA | Current Comand Valve-Brake |
| 00 06 | Wireless ID【RFBUSID】 | R/M | 1 | 0x01-0x7C | Suggest not use function code |
| 00 08 | Valve restoration after power-cut | R/M | 1 | specific data as follows: | |
| | | | | 0x01 | Power-on valve-on automatically |
| | | | | 0x07 | Power-on valve-off automatically |
| | | | | 0x09 | Power-on no operate |
| | | | | 0x43 | Power-on act as previous action |
| 00 09 | Modbus command delay return | R/M | 1 | 20-500ms | Return ofter delay xxMs of receive Modbus |
| 00 0D-00 0E | Production Date | R | 2 | Sequence：0xYHYL 0xMMDD,14 0D 04 1C ==>April 28th,2013 | |
| 00 11-00 13 | Use times | R | 2 | Sequence：0x00HH 0xMMLL,Used cycles=LL+256*MM+65535*HH | |
| 00 14 | Actuator current position | R | 2 | 0xHHLL: Open Angle Value = (A/10)% A=HH*256+LL | |
| | | | | | |
| 00 23 | Communicate frequency[Freq_Index] | R/W/M | 1 | 0-31 | [862+Freq_Index]MHZ |
| 00 24 | Spread factor[SpreadFactor_Index] | R/W/M | 1 | 0-6 | SpreadFactor=6+SpreadFactor_Index |
| 00 25 | Send out Power[PowerBuf] | R/W/M | 1 | 0-3 | 0:20db 1:17db 2:13db 3:10db |
| 00 26 | Frequency deviation[FdevBuf] | R/W/M | 1 | 35KHZ | |
| 00 27 | Buffer bandwidth[BandBuf] | R/W/M | 1 | 0-9 | |
| | Set value( [0] [1] [2] [3] [4] [5] [6] [7] [8] [9]) | | | | |
| | corresponding bandwidth ([7.8][10.4][15.6][20.8][31.2][41.7][62.5][125][250][500]) | | | | |
| 00 29 | Low speed optimization[LowDataRateOptimize] | R/W/M | 1 | 0/1 | 0: Close 1：Open |
| 00 2C | Lead code length[Length_LeadCode] | R/W/M | 1 | 4-15 | Default is 8 |
| 00 2D | Broadcast reply sequence[Idx_BroadReply] | R/W/M | 1 | 1-0x7C | receive retruned broadcast sequece |
| 00 2E | Broadcast delay reply[DelayMs_BroadReply] | R/W/M | 1 | 20-900ms | return after delay xxmsreceive broadcast |
| R:support long-distence read W:support long-distance write M:revise by menu | | | | | |

**【Communicational function code between host and RS485 bus valve】**

0x03/04 Read data out of valve register, even several continuous data is permitted.

0x06 Write single byte data in valve register：xx xx double byte data

**【 Command formats of reading valve operation sent by host 】**

| Valve_Addr | 03/04 | 00 | XX | 00 | YY | CRCL | CRCH |
|---|---|---|---|---|---|---|---|

Within：Valve_Addr means valve bus address of reading operation；

03/04 means reading valve register；

00 XX means initial address to be read out(XX is only a 19-byte address from 0x01 to 0x13)

00 YY means (YY-XX+1)data reading from 00 XX to(00 XX+00 YY) in register；

CRCL CRCH means two-byte CRC check code sent by host

**【 Data formats of reading valve to operate valve returning accurately 】**

| Valve_Addr | 03/04 | YY*2 | $DAT_1$ | $DAT_2$ | … | $DAT_{YY*2}$ | CRCL | CRCH |
|---|---|---|---|---|---|---|---|---|

Within：Valve_Addr means returning data of valve bus address after receive reading operation；

03/04 means reading operation of valve register ；

YY means having read out YY*2 byte data；

$DAT_1$ $DAT_2$…$DAT_{YY}$ are read-out data；

CRCL CRCH means two-byte CRC check code calculated by valve MCU.

**【 Command formats of single-byte writting valve operation sent by host 】**

| Valve_Addr | 06 | 00 | XX | 00 | DAT | CRCL | CRCH |
|---|---|---|---|---|---|---|---|

Within：Valve_Addr means valve bus address to be written；

06 means writting valve register；

00 XX means register address in need of writting in register(only writtable register is permitted) ；

00 DAT is write-in data in demand；

CRCL CRCH means two-byte CRC check code calculated by host.

**【 Data formats of valve returning on inaccurate operation 】**

| Valve_Addr | 06 | 00 | XX | 00 | DAT | CRCL | CRCH |
|---|---|---|---|---|---|---|---|

Within：Valve_Addr means returnig data of valve bus address after receive writtng operation ；

06 means writting valve register operation；

01 is fixed value which means having wrote a byte in valve register；

Copy the returned data,means operate correctly

【 Data formats of valve returning on inaccurate operation 】

| Valve_Addr | Function code+0x80 | Returned code | | CRCL | CRCH |
|---|---|---|---|---|---|

Within：Valve_Addr means valve bus address on error operation；

Function code+0x80 means inaccurate return；returned code:details as follow：

0x00 //No error
0x65 //101 //slave station not support this function code
0x66 //102 //slave station not support this address range
0x67 //103 //slave station not support this data range
0x68 //104 //slave station device error
0xC9 //201 //ON/OFF command code:slave station not support
0xCA //202 //slave station forbid long-range revise wireless data
0xCB //203 //illegal Modbus data
0xCC //204 //CRC checkout error

## II Read out valve data which are written in register

1、Command formats of reading out single byte data

Send format：Valve_Addr 03/04 00 XX 00 01 CRCL CRCH

Within：Valve_Addr means valve bus address on present operation；

03/04 means reading valve register operation；

00 XX means register address to be read out (XX could only be0x01-0x13)；

00 01 means having read out a byte data；

CRCL CRCH means two-byte CRC check code.

When Valve_Addr =0，null data return；

When Valve_Addr!=0，valve corresponded in address code could return；

When the returning code is correct,it will get：Valve_Addr 04 02 DATH DATL CRCL CRCH

Within：Valve_Addr means returning data of valve bus address after receive reading operation；

DATH,DATL are read-out data；

CRCL CRCH means two-byte CRC check code calculated by valve MCU.

【Example】Query valve bus addr. 01 state ：

Host sends：01 04 00 01 00 01 60 0A

Valve returns：01 04 02 00 AC b9 4d (Bus addr.is 01 valve is in open position)

01 04 02 00 BC b8 81 (Bus addr.is 01 valve is in close position)

III Revise the input register data in valve

Only could revise and write into the writeable register

1.Control valve action

Send format:Valve_Addr 06 00 04 00 XX CRCL CRCH

Within：Valve_Addr means valve bus address on present operation；

00 04 means register of saving valve control command；

00 XX means control command code,detail as follows；

CRCL CRCH means two-byte CRC check code

When Valve_Addr =0，no data returned；

When Valve_Addr!=0，fittabel address number valve returned；

If return right,valve returned：Valve_Addr 06 00 04 00 XX CRCL CRCH

【Example】Bus addr.01 valve controlling command as follows：

| Control command | Send command | valve return | remarks |
|---|---|---|---|
| valve-off | 01 06 00 04 00 00 C8 0B | 01 06 00 04 00 00 C8 0B | |
| valve-on | 01 06 00 04 00 64 C9 E0 | 01 06 00 04 00 64 C9 E0 | |
| open30° | 01 06 00 04 00 1E 48 03 | 01 06 00 04 00 1E 48 03 | 4-position 2-way specific |
| open60° | 01 06 00 04 00 3C C8 1A | 01 06 00 04 00 3C C8 1A | 4-position 2-way specific |
| open to B33 | 01 06 00 04 00 B3 89 BE | 01 06 00 04 00 B3 89 BE | 3-position 3-way specific |
| brake | 01 06 00 04 00 BA 49 B8 | 01 06 00 04 00 BA 49 B8 | |

2.Broadcast command

Special functional code list[TSFunCode]

0xFD //Broadcast address,every address could receive this command,but not return info.

0xE6 //reply to broadcast command

0xE4 //reply to broadcast read command

[1] Broadcast command means the command of ID=0xFD Function AddrH AddrL DataH DataL CRCH CRCL

ID is different with normal communication command,ID=0xFD.

[2] Broadcast command only could read “operate state”and “valve control command”,details as follows:

#valve-on command: FD 06 00 04 00 64 dd dc

#valve-off command:FD 06 00 04 00 00 dc 37

#read state: FD 04 00 01 00 01 74 36

[3] Broadcast return format:

Reply to write command : 0xE6 RFBUSID [ReplyCode] CRCH CRCL

Reply to read command : 0xE4 RFBUSID [Workstatus] CRCH CRCL

# RFBUSID: means valve wireless ID code,

# [ReplyCode]： means whether valve act broadcast command or not,details refer to ahead return code list

# [Workstatus]： details refer to actuator operation state list

[3] In condition of same frequency band and same wireless data,every valve would reply the received command.

[4] Valve delay after recive wireless command:After(Idx_BroadReply*DelayMs_BroadReply)ms,broadcast command return in turn.

[5] In actual use,the valve in need of action could be set in one special communication frequency.The valve in need of sperate control could set in another communication frequency.

[6] If it couldn't receive signal in returning progress,that could increase DelayMs_BroadReply time by menu.

3. Wireless data detail instruction

[1] Communication frequency [Freq_Index]：862M-893M could be set by menu.

[2]Spread factor[SpreadFactor_Index]：Max spread factor=12,the value larger,the communicate distance longer,but when the communication speed is slower,the spread factor=12,buffer bandwidth only could be set 62.5K,125K, 500K,or it couldn't intercommunicate.

[3] Transmitted power[PowerBuf]：recommend to set 20db in max.

[4] Frequency deviation[FdevBuf]：forbid to revise,the defaults is 35KHZ.

[5] Buffer bandwidth[BandBuf]: co-use with spread factor,the larger value is,the faster speed is.According to physical in fact,set as larger value as possible to improve speed.

[6] Low speed optimization[LowDataRateOptimize]：When spread factor≥11,pls open low speed optimization,or the communication error rate would be high enough to affect communication distance.

[7] Length of lead code[Length_LeadCode]：lead code is used to improve communiation data reliability,the default value is 8 which could be increase propely,if failed code rate is high.

[8] When Modbus renturn delay：means delay AckDelayMs receive normal Modbus communication command,then return data. Since it has delayed time from emitter convert to receiver,which could ensure the emitter could receive valve returned data .Pls set value as small as possible in order to ensure normal communication.

[9] Brooadcast command return sequence Idx: the broadcast commmand sequence only has effect to broadcast sequence commmand

[10] Broadcast command return delay:the broadcast commmand delay returned only has effect to broadcast command.

Usage instruction:

[1] Ensure the 5 data communicating needed are consistent: 【communication frequency】【spreading factor】【buffer bandwidth】【Low speed optimization】【lead code length】,or it couldn't communicate normally.

[2] The defaults is antenna in the condition of open square,test communicate distance≥2K.m,but in actual use,since the affect of environment,communicate distance might has slight deviation.So wireless control is suggested to install in the high place, take advance of enlarge communication distance,improve the communication data speed.

[3] In actual use,pls set the buffer bandwidth 500K,then test if it could communicate normally,if not,decrease the buffer bandwidth,so that to improve communicate speed rate as high as possible.

[4] The lower communicate speed is,the larger value broadcast delay is:From our test spread factor=12,buffer bandwidth= 125KHZ,so it would be suitable to set the broadcast command delay 250MS.If spread factor=12,buffer bandwidth= 500KHZ, it would be suitable to set broadcast command delay 100ms.Details need refer to actual test,the data is only a reference.