WAC2
USER MANUAL

IoT Wiegand Access Controller

29th June, 2018

AVEA International Company Limited

(http://avea.cc)

Table of Contents

WAC2 IoT Access Controller

WAC2 is an access controller which embedded with Ethernet connection and using HTTP for communication control protocol. Hence, it can communicate directly with Apache Web Server or Microsoft Internet Information Server. WAC2 provides 3,000 access cards and 50,000 access logs records.

It is implemented as a HTTP web client. Various page extension is selectable, like .php, .asp, .cfm, .pl, .htm, .html and .aspx. So various HTTP server systems (e.g. IIS and ASP from Microsoft; Apache with PHP enabled and mySQL database server from Unix/Linux), a complete and powerful time attendance system can easily be implemented.

When cards are presented, the transaction information is stored inside the memory of the reader with 50,000 records allowed. The record is looked up from the local access table with 3,000 records allowed. If access allowed, it will engaged the relay to allow door open. The memory is retained even power is lost.

The controller provides two Wiegand RFID reader interface and two relays output for controlling door access.

Whenever there is network connection and the server is running, those records will be sent to the server. Upon confirmation by the server, the controller will be told to delete the records accordingly.

1. Setup the reader

In order to work properly, WAC2 must be configured correctly before putting into actual usage. For normal condition, the blue LED is blinking.

1.1 Default Settings

The default configurations of the WAC2 is listed as follows:

Parameter	Description	Default
IP	IP address of the reader itself	192.168.1.234
GW	Gateway IP address	192.168.1.1
NM	Netmask	255.255.255.0
WS	HTTP server IP address	192.168.1.1
PT	HTTP server port number	80
EXT	Page extension	php

1.2 Reset configurations to default

If for any reasons, the state of the WAC2 is not known. Before power up, set DIPSW 1 to on position. When power on, the settings of WAC2 will be cleared and set to the default settings. DIPSW 1 must be set to off for normal operation.

1.3 Force DHCP mode

When DIPSW 2 is set to on position. DHCP mode will be forced to enable for automatic configuration of the WAC2. A DHCP server must be available in the network for this operation. If the DHCP server configured with the options www-server, i.e. option 72, the web server address of will also be set automatically.

Example of dhcpd.conf:

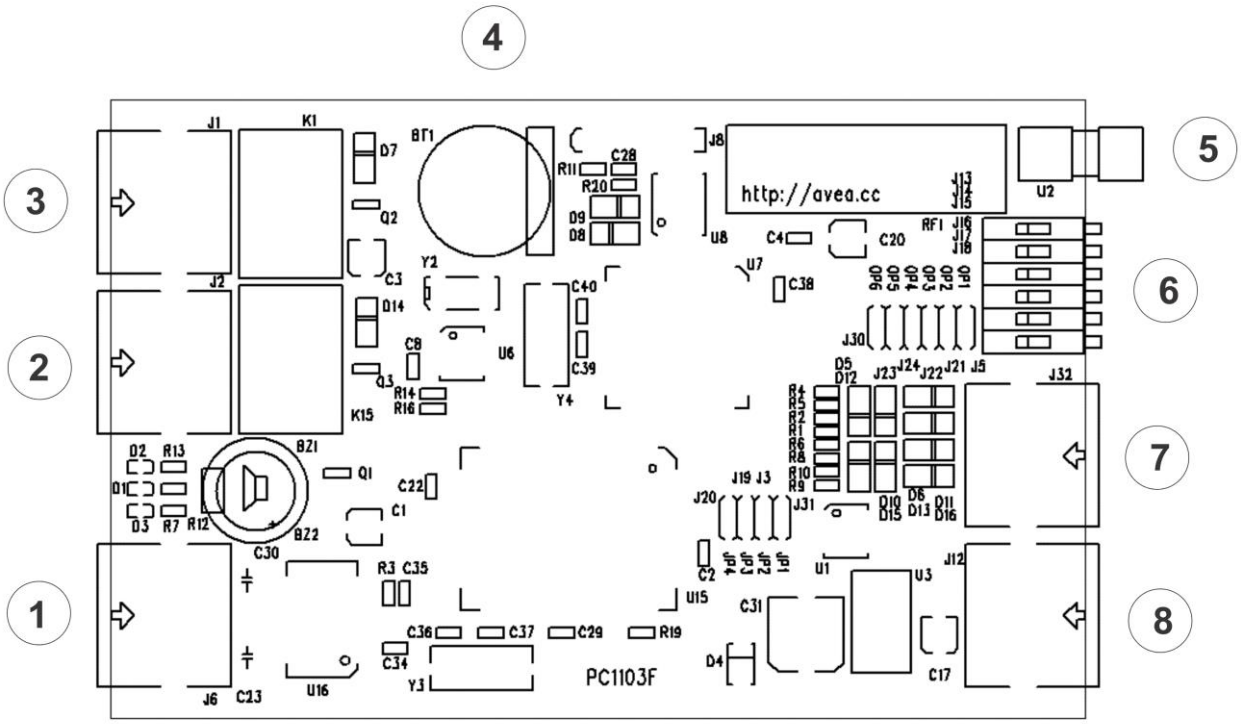
```
#example of dhcpd.conf for AVEA's IoT devices
ddns-update-style   ad-hoc;

subnet 192.168.1.0 netmask 255.255.255.0 {
    # --- default gateway
    option routers 192.168.1.1; # default gateway
    option subnet-mask 255.255.255.0;   # netmask
    option www-server 192.168.1.123;    # it must setup to the web server's IP
    range dynamic-bootp 192.168.1.10 192.168.1.99;
    default-lease-time 300;
    max-lease-time 3600;
}
```

2. Installation

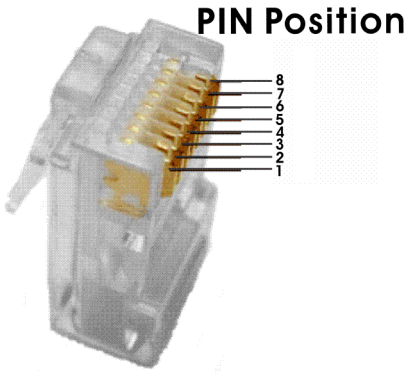
In order to work properly, WAC2 must have stable power supply, a 10-baseT compatible Ethernet connection. The WAC2 provides interface to connect two Wiegand 26 or Wiegand 34 RFID reader and each associated with its own relay control.

Remember to remove power when doing any installation or connection.

The following diagram show the PCB of the WAC2 controller.



All connections from the WAC2 is through the RJ45 connector. The following diagram show the pinouts of the RJ45 plug.



2.1 Power Input and Ethernet Connection

Circle 1, is a RJ45 terminal which is the connection to Ethernet network and also the input power to the controller.

The pinouts of the terminals is as follows:

PINOUT	DESCRIPTIONS
1	ETHERNET, TX+
2	ETHERNET, TX-
3	ETHERNET, RX+
4	POWER
5	POWER
6	ETHERNET, Rx-
7	GROUND
8	GROUND

WAC2 requires 9 to 12V DC 500mA for normal operation. Since the POWER/GROUND will be connected to the Wiegand port as well, more current will be needed to drive them.

The power can be connected by the POE-02 power injector or equivalent.

2.2 Wiegand Interface

Circle 7, is a RJ45 terminal for connecting Wiegand 26/34 RFID reader.

The pinouts of the terminals is as follows:

PINOUT	DESCRIPTIONS
1	POWER
2	CHANNEL 1, DATA0
3	GROUND
4	CHANNEL 1, DATA1
5	POWER
6	CHANNEL 2, DATA0
7	GROUND
8	CHANNEL 2, DATA1

The POWER/GROUNG pair is the power supply to the controller which can be used to power the Wiegand readers. Be careful of the correct voltage ratings.

DATA0/DATA1 corresponding to the Wiegand's DATA0 and DATA1.

2.3 Relay Control

Circle 2, is a RJ45 terminal which is the channel 1 relay output which associate with channel 1 wiegand reader.

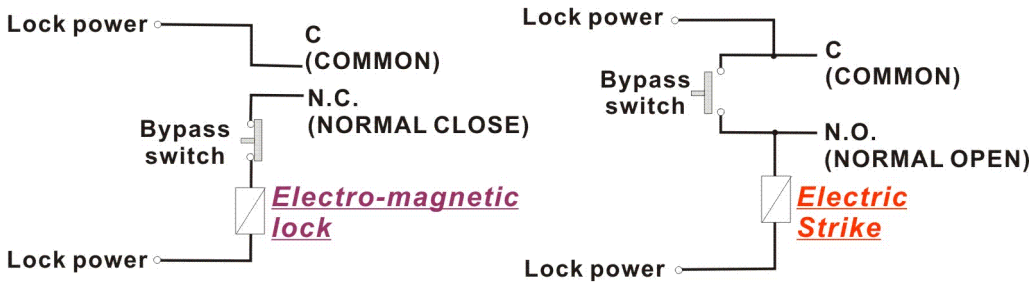
Circle 3, is a RJ45 terminal which is the channel 2 relay output which associate with channel 2 wiegand reader.

The pinouts of the terminals is as follows:

PINOUT	DESCRIPTIONS
1	NC
2	NC
3	CO
4	CO
5	CO

6	CO
7	NO
8	NO

NC - normal close, it is connected to CO normally.
CO - common, it will connects to either NO or NC, but not both.
NO - normal open, it is open circuit normally.



The maximum switching voltage of the relay is 120VAC or 60VDC, and the maximum switching current is 2A.But it is recommend to drive low DC voltage only because of the RJ45 connector's limitation. Overloading the maximum ratings may cause damages to the reader permanently and cannot be repaired anymore.

A bypass switch can be added for the manual control of an electric strike or magnetic lock for door access.

2.4 DIPSW Options

Circle 6, is the DIPSW for selecting varies options for the WAC2.

2.4.1 Reset to default

By setting DIPSW position 1 to on, the controller configuration will be reset to the default state. It must be set to off position for normal operation.

2.4.2 Force DHCP enable

By setting DIPSW position 2 to on, the controller configuration will be forced to enable DHCP for IP assignment.

It assumes a DHCP server is available in the local network.

2.4.3 Test mode

By setting DIPSW position 3 to on, the controller configuration will be set to the access the AVEA's server for testing purpose. It must be set to off position for normal operation.

It assumes a DHCP server is available in the local network.

Setting DIPSW 1 will clear this settings.

2.4.4 Reserved DIPSW

DIPSW position 4, 5 and 6 is reserved for future use.

2.5 Battery

Circle 4, a CR1220 lithium cell battery is required for the normal operation of the real-time clock. It is not included in the package. Customer needs to purchase by themselves. The battery in installed in Circle 4.

2.6 Reserved

Circle 5 and Circle 8 is reserved.

2.7 Mechanical Dimensions

Height: 25mm
Width: 78mm
Depth: 112mm

3. Software

WAC2 is a HTTP client with PHP enabled. It will send PHP URL requests to the HTTP server with the various parameters. And it accepts responses from the HTTP server to perform various actions on the reader.

3.1 Requests to HTTP server

The followings are the variables presented to the PHP. User can access it correspondingly.

3.1.1 \avea.php

The reader send a GET request to access a fixed web page of the HTTP server: avea.php located on the root of the website. The file extension is user selectable, e.g. avea.asp, avea.cfm avea.pl, avea.htm and avea.html. But it is server changeable to any numeric filename located on the root.

3.1.2 $date and $time

Hold the date and time of that request. The format of date is YYYY/MM/DD, e.g. 2018/06/29 means 29th June 2018. The 24 hour format of time is HH:MM:SS where HH is hour, MM is minute and SS is second.

3.1.3 $id

It is the IP address of the reader, e.g. 192.168.1.234. It must be set to different value for each reader in the same subnet in order to function properly.

3.1.4 $uid

It is the unique code number of the RFID card presented to the reader in hexadecimal format.

3.1.5 $cmd

It holds the action that was taken on the reader.

I. $cmd=PU, it is sent once after the reader is just power up.

II. $cmd=HB, it is sent when the programmed heartbeat rate is reach. The default heartbeat rate is 300 seconds.

III. $cmd=PG, it is sent when the reader is being pinged. It can be used to initiate event from the server side.

IV. $cmd=REC, it is sent with an ID card is read on the reader. The card number is stored in $uid.

V. $cmd=ET, when changing access card operation is taken. It is sent after the operation is completed. So the server can decided to take further actions.

3.1.6 $channel

It signify the action is coming from which channel of the WAC2.

I. $channel=1, the action is coming from channel 1

II. $channel=2, the action is coming from channel 2.

3.1.7 $mode

For WAC2, $mode=AC1. So it can be used to identify the reader.

3.1.8 $ver

It is the firmware revision. Currently it is 1.39.

3.1.9 $recno

Hold the reference number of the record in decimal format. The server must response this parameter in order to acknowledge the record and confirm the deletion of that record.

3.1.10 $rdate and $rtime

Hold the date and time of the stored record. The format of date is YYYY/MM/DD, e.g. 2018/06/29 means 29th June 2018. The 24 hour format of time is HH:MM:SS where HH is hour, MM is minute and SS is second.

3.1.11 $sid

This is the value set by the SID response.

3.1.12 $deviceid

This is the unique id of the reader.

3.1.13 $md5

It is the MD5 checksum of the user's MD5 secret key, date and time. It will be sent together in the request for identification purpose.
For examples:

assuming

- MD5 secret key is '01234567' (ASCII),
- date of the request is '2012/08/08' (ASCII),
- time of the request is '10:19:54' (ASCII)

the combined string is

- '012345672012/08/08-10:19:54' (ASCII)

hence the MD5 checksum of the combined string is

- b62a8cf4adfdd10874f1121686b0bba9

On the server side, the MD5 secret key is known, so it can compute the MD5 checksum and check against the received checksum for verification.

Since the MD5 secret key is unknown to others, only those authenticated reader can sent out correct MD5 checksum.

3.1.14 $event and $cn

They are valid for the $cmd=ET. Whenever add/append/delete card or erase access table is responded by the web server, the controller finished the operation and will send a request $cmd=ET to the web server with those parameters.

- $event valid values are { ‘add’ , ’app’ , ’del’ , ’clr’ , ’err’ }
- $cnd is the RFID card number

3.2 Responses from HTTP server

After sending request, WAC2 will wait for a response from the HTTP server. It looks for the starting flag <AVEA> and the ending flag </AVEA>. Then it interprets the strings in between. There must be no space between the keywords and parameters. The maximum size of the response packet should be less than 600 bytes.

The following responses are supported:

3.2.1 HB=xxxx

Set the heartbeat rate of the reader in seconds. It is a fixed length format in decimal value. Example:

HB=0300 set the reader send a heartbeat request to the HTTP server once in 300 seconds.
HB=0000 to disable the heartbeat request.

3.2.2 BEEP=x

Make a beep sound on the reader. It is a fixed length format. Example:

BEEP=1 make a short beep.
BEEP=0 make a long beep.

3.2.3 CK=YYYY-MM-DD HH:MM:SS

Set the clock of the reader. The year must be in 2000 to 2099. It is a fixed length format in decimal values. Example:

CK=2007-01-23 12:34:56 set the clock to 23rd Jan 2007, pm12:34:56.

3.2.4 GRNT=xx

Set the reader to a grant access state, i.e. to engage the relay from NO state to NC state for xx seconds and return to NO state. Meanwhile a LED will be turn on and off simultaneously. It is a fixed length format in decimal value. Example:

> GRNT=03 set the relay to NC state and LED on for three seconds and return to NO state and LED off.

3.2.5 DENY

Set the reader to a deny access state, i.e. to make sure the relay is in NO state.

3.2.6 ROOT=xxxxxxxx

This will change the root page to be access by the reader. It is a fixed length format in decimal value. For example: ROOT=00024689, this will make the reader to access the page /24689.php rather than the default /avea.php. By setting ROOT=00000000 will reset to access the default page.

3.2.7 EXT=x

This will change the root page's file extension to be access by the reader. It is a fixed length format.

Value of x	File extension
0	.php
1	.asp
2	.cfm
3	.pl
4	.htm
5	.html
6	.aspx

3.2.8 DHCP=x

This will control DHCP feature of the reader. If DHCP=1, DHCP is enabled. If DHCP=0, DHCP is disabled. If enabled, it will send requests to DHCP server to acquire the following items:

- host IP
- netmask
- default gateway
- www-server IP

3.2.9 IP=xxx.xxx.xxx.xxx

This is a fixed length command and values are in decimal. It will change the IP address of the reader. Example:

IP=192.168.001.234

3.2.10 GW=xxx.xxx.xxx.xxx

This is a fixed length command and values are in decimal. It will change the default router of the reader. Example:

GW=192.168.001.002

3.2.11 NM=xxx.xxx.xxx.xxx

This is a fixed length command and values are in decimal. It will change the netmask of the reader. Example:

NM=255.255.255.000

3.2.12 WS=xxx.xxx.xxx.xxx

This is a fixed length command and values are in decimal. It will change the web server IP address to be accessed by the reader. Example:

WS=192.168.001.001

3.2.13 PT=xxxxx

This is a fixed length command and values are in decimal. It will change the port number to be used to access the web server in decimal. Example:

PT=00080

3.2.14 RLY=x

This is a fixed length command and values are in decimal. If RLY=1, the relay will be turned on. If RLY=0, the relay will be turned off. The state of relay will be affected by the subsequence commands.

3.2.15 REC_DEL=xxxx

This is a fixed length command and xxxx is a 16bit hexadecimal value. This will acknowledge the reception of the record. Once the reader get this response, it will delete that record accordingly and cannot be recovered.

3.2.16 SID=xxxxxxxx

This is a fixed length command and values are in hexadecimal value. It is a non-volatile value and will not be changed after power removed.

3.2.17 AC=xxxxxxxxxx

Add access allowed card number to local access table. This is a fixed length command. xxxxxxxxxx is 10 decimal digit of the access card number. It is a non-volatile value and will not be changed after power removed.

The controller will search for the local access table to check whether the card is in the list. So it will take time for the searching.

A request $cmd=ET will send to the web server to signify the finished of the command:

- $event=add for success
- $event=err for error
- $cn is the corresponding card number

3.2.18 DC=xxxxxxxxxx

Delete access allowed card number to local access table. This is a fixed length command. xxxxxxxxxx is 10 decimal digit of the access card number. It is a non-volatile value and will not be changed after power removed.

The controller will search for the local access table to check whether the card is in the list and delete it accordingly.

A request $cmd=ET will send to the web server to signify the finished of the command:

- $event=del for success
- $event=err for error
- $cn is the corresponding card number

3.2.19 APC=xxxxxxxxxx

Append access allowed card number to local access table. This is a fixed length command. xxxxxxxxxx is 10 decimal digit of the access card number. It is a non-volatile value and will not changed after power removed.

The controller will not search for the duplication of the card, hence, it will take short time for the appending.

A request $cmd=ET will send to the web server to signify the finished of the command:

- $event=app for success
- $event=err for error
- $cn is the corresponding card number

3.2.20 CLEAR=ALL

This command will clear the local access table completely and cannot be recovered. It will take relatively long time to complete the action.

A request $cmd=ET will send to the web server to signify the finished of the command:

- $event=app for success
- $event=err for error

3.2.21 ACPIN=xxxx

This allows the PIN only access to the controller. This is a fixed length command. xxxx is 4 decimal digit of the access PIN number. By pressing the PIN number and then '#' key, the PIN entered will be checked. If it equals xxxx, access will be granted to open door. Setting ACPIN=0000 will disable the PIN only access.

3.2.22 PPC=x

PPC stands for Pin plus Card access. This is a fixed length command. x is a decimal digit of either 0 or 1.

If PPC=0, card only access is enabled. Presenting the granted card will allow access.

If PPC=1, PIN and CARD mode is enabled. User need to enter the PIN number then presenting the granted card to get access.

4. PHP Code Example

The following is an self explainable example of PHP script located on the HTTP server. The filename is \avea.php.

```
<html>
<body>
<?php
// setup variables
$cmd=$_GET[“cmd”];
$mode=$_GET[“mode”];

$now=time();   // stamp the current time
$st=date('Y-m-d H:i:s',$now); // set the datetime string to  correct format
$mycard=359452;     // replaced by your card number
$rtime=$date . $time;    // access the date and time of the reader
$remote_open=1;

echo "<AVEA>"; // starting flag
switch ($cmd) {
case "PU":     // power up
     echo "CK=$st"; // set clock
     if ($mode==ID2) {   // for web08s only
          echo “DHCP=1”;      // turn on the DHCP feature
     }
     break;

case "REC":    // stored record
     $code=$_GET[“code”];
     $rectime=$_GET[“rtime”];
     $recdate=$_GET[“rdate”];
     //
     // you can write you own code here, i.e. store the record into database
     //

     // the following will acknowledge the reception of the record
     echo "REC_DEL="; printf("%04X",$_GET["recno"]);
     break;
```

```
case "HB":     // heartbeat
     echo "CK=$st"; // set clock
     break;

case "PG":     // being pinged
     echo "CK=$st"; // set clock
     break;
}
echo "</AVEA>";     // ending flag
?>
</body>
</html>
```