AGD | SAFER GREENER MORE EFFICIENT

TECHNICAL ADVICE NOTE

AGD650 Ethernet Variants

TAN50

AGD Systems Limited
White Lion House, Gloucester Road,
Cheltenham, GL51 0TF, UK

CONTENTS

1. INTRODUCTION

AGD650 Ethernet variants are identified by product numbers in the format 650-40x-722. These models retain the full functionality of the non-Ethernet variants, including dual opto-isolated outputs and Wi-Fi connectivity.
In addition to these features, Ethernet-enabled AGD650 units include an Ethernet port, which allows:

- Access to the web interface for configuration and management.
- Retrieval of frame-by-frame zone occupancy data.

This document outlines the additional capabilities available in Ethernet-enabled AGD650 models.

Security Considerations

The AGD650's Ethernet connectivity is designed for local networks only and must not be connected to the internet or deployed on public or enterprise networks.
Recommended precautions include:

- Network segmentation - to isolate the device from critical systems.
- Firewall rules - to restrict external access.
- Access controls - to prevent unauthorised configuration changes.

Failure to implement appropriate security measures MAY leave AGD650 and/or the wider system vulnerable to unauthorised access, data breaches or enterprise-scale cyber-attacks.

2. ETHERNET CABLE INTERFACE

The AGD650 features a Category 6 (Cat 6) Ethernet interface accessible via the rear product connector. This interface supports both Power over Ethernet (PoE) and non-PoE operation.

- Where PoE is used, the AGD650 receives receives power and transmits data through the Ethernet connection.
- Where PoE is not used, the unit requires an alternative power source. This is provided via the 'low voltage' cable/connector, which also accommodates the two opto-isolated outputs.

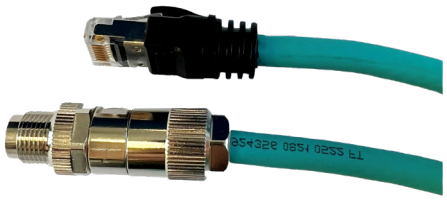
Mating Connector Details

The AGD650 is compatible with a field-installable cable-mount connector:

- Connector Model: Binder 99-3787-810-08
- Wiring: The connector should be wired in accordance with the manufacturer's instructions to ensure correct functionality and reliable operation.

Pre-Assembled Cable Option

- Alternatively, a pre-assembled cable is available in various lengths from Stewart Connector (or equivalent) that provides a convenient, ready-to-use solution.



COMPATIBLE CABLES FROM STEWART CONNECTOR	
PN	**LENGTH**
BM-MARM003F	3ft
BM-MARM007F	7ft
BM-MARM015F	15ft
BM-MARM030F	30ft

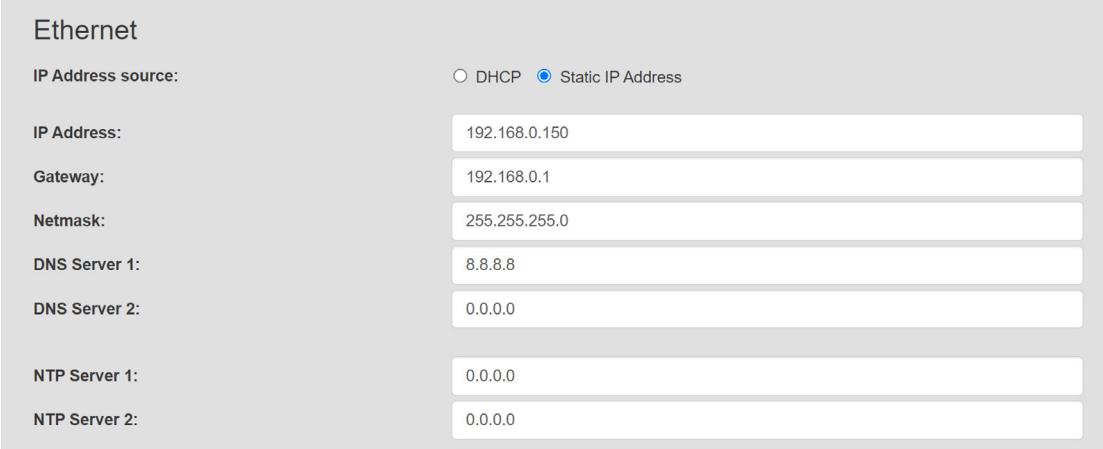
P.O.E - POWER OVER ETHERNET REQUIREMENTS

AGD650 P.O.E REQUIREMENTS
Type 1 Class 3 Device
802.3at Compliant and Isolated
6.49W to 13W (10.5W nominal)
37.0v to 57.0v
Supports Mode A and Mode B Wiring

3. NETWORK CONFIGURATION

The AGD650 can be connected to a network and obtain an IP address via a DHCP server. Alternatively, the IP address, gateway, and subnet mask can be manually configured to customer-specified values.

- Configuration Interface: Ethernet settings can be modified from the Site/Network Settings tab on the SETUP page.
- Default Settings: The factory-default Ethernet settings are displayed in the screenshot below.
- Wi-Fi Access: If you are unable to connect using the default ethernet settings, please refer to the product manual for instructions on connecting via Wi-Fi to update the configuration.



Important Notes:

- After modifying the Ethernet settings and clicking Next, the Ethernet connection will reset to apply the changes.
- If you are connected via Ethernet and the IP address is changed, you must reconnect using the new IP address.

DNS and NTP Configuration

DNS Servers:

- The AGD650 supports configuration of up to two DNS servers.
- To unset a DNS server, clear the field so that it is replaced with all zeros.
- If no DNS servers are set, DNS resolution will not be available.

NTP Servers:

- The AGD650 allows configuration of up to two NTP servers.
- To unset an NTP server, clear the field so that it is replaced with all zeros.
- If no NTP servers are set, the AGD650 will attempt to locate an NTP server via a network connected device, but this requires a configured and accessible DNS server.

Time Synchronisation Behaviour

If a NTP server is unavailable, the AGD650 will synchronise its date and time from the connected device during initial setup.

- If the unit is power-cycled without an available NTP server, it will not retain the current date and time.
- While this does not affect core functionality, the timestamp in the JSON output message (see Section 4) will be incorrect.
- AGD650 has fallback NTP servers preprogrammed, that it will attempt to utilise should no network connected NTP server be specified.

4. CONNECTING TO THE WEB INTERFACE VIA ETHERNET

To access the AGD650 web interface from a computer connected via Ethernet, enter the following URL in the address bar of an internet browser:

<Ethernet IP Address>:8080

Where <Ethernet IP Address> is replaced with the DHCP/static IP address of the AGD650.

- For static IP configurations, the assigned IP address is displayed on the Site/Network Settings tab of the SETUP page.
- For DHCP-assigned IP addresses, the current address can be found by logging into the admin page of the network router.

5. DATA OUTPUT VIA ETHERNET

The AGD650 outputs data in JSON format for each image frame that is analysed. Therefore the data output rate is the same as the detection frame rate (approximately 6.7 fps).

5.1 Data output format

An example of the JSON output is shown below:


```
{
  “SN”: “000000-0000”,
  “msUTC”: 1739451919555,
  “version”: 2,
  “zones”: [
    {
      “index”: 1,
      “roadusers”: [
        {
          “H”: 0.7204192280769348,
          “W”: 0.4059552550315857,
          “X”: 0.5816567540168762,
          “Y”: 0.4653209447860718,
          “className”: “car”,
          “trackID”: 0
        }
      ],
      “state”: 1
    },
    {
      “index”: 2,
      “roadusers”: [],
      “state”: 2
    }
  ]
}
```

The fields in this data have the following meanings:

- SN: The serial number of the AGD650 that is outputting the data.
- msUTC: A timestamp expressed as milliseconds since 1st January 1970 UTC.
- version: A version number for the JSON format.
- zones: An array containing one entry for each zone.
- index: The index of the zone (starting from one).
- state: The current state of the zone where: 0 = not set, 1 = in detect, 2 = in non-detect, 3 = timed out (see the product manual for an explanation of the timeout feature).
- roadusers: An array of road-users currently in the zone. This will only include those road-user classes that are currently set to trigger the zone.
- className: The type of road-user.
- trackID: A 32 bit unsigned integer that is assigned to each road-user by a tracking algorithm. This can be used to identify the same road-user across multiple frames so that count data can be obtained. Track IDs begin at 0 when the unit is powered on and are incremented by 1 for each road-user identified.
- H: The height of the road-user's bounding box, relative to image height.
- W: The width of the road-user's bounding box, relative to image width.
- X: The horizontal position of the centre of the road-user's bounding box, where 0 corresponds to the left side of the image and 1 corresponds to the right side of the image.
- Y: The vertical position of the centre of the road-user's bounding box, where 0 corresponds to the top of the image and 1 corresponds to the bottom of the image.

5.2 Receiving the data

The JSON data can only be received via an Ethernet connection (not over WiFi). In order to receive the data, the connected computer/controller needs to have the ZeroMQ library (https://zeromq.org/) installed and to be running software that uses the ZeroMQ library to receive the data. This software can be written in a range of different languages including C, C++, C#, Java and Python. The following Python code shows an example of how to receive data from the AGD650 (where <ipaddress> is the current Ethernet IP address of the AGD650):

```
import zmq

context = zmq.Context()
sock = context.socket(zmq.SUB)
sock.setsockopt(zmq.SUBSCRIBE, b'') # note: this line contains two single quote marks
sock.connect("tcp://<ipaddress>:9003") # note: this line contains two double quote marks
while True:
    message = sock.recv()
    print(message)
```

6. CONFIGURING DETECTION ZONES TO OBTAIN ACCURATE COUNT DATA

Obtaining count data with the AGD650 relies on an algorithm which tracks objects from frame to frame. If an object is not detected for several frames (e.g. due to being occluded) and then reappears, this may lead to the tracking algorithm deciding that a new object has entered the field of view, resulting in over-counting. Over-counting is therefore most likely to occur in detection zones that get very busy and contain a large number of road users that tend to occlude one another (e.g. pedestrians, cyclists, e-scooter riders). In very busy areas, when a zone is to be used primarily for obtaining count data, it may be better to locate the zone at a point where road-users will travel through the zone, rather than stop within it. Attention also needs to be paid to siting the AGD650 so that smaller road-users, e.g. cyclists and scooter riders, will not be occluded by larger vehicles.

SAFER
GREENER
MORE EFFICIENT

AGD Systems Limited
White Lion House, Gloucester Road,
Cheltenham, GL51 0TF, UK

Tel: +44 (0) 1452 854212
Email: info@agd-systems.com
Web: agd-systems.com