

# Application Interface Eagle Area Monitor®

---

|           |                    |
|-----------|--------------------|
| System    | Area Monitor®      |
| Version   | 0.3                |
| Author(s) | Julie Vandenabeele |
| Date      | 25-04-2012         |
| Status    | Draft              |

---

**Eagle Vision Systems B.V.**

Energiestraat 16B  
1411 AT Naarden  
Tel: 035-6952818  
Fax: 035-6953842  
Email: [info@eaglevision.nl](mailto:info@eaglevision.nl)

**Copyright © Eagle Vision Systems B.V., 2010-2011**

Alle rechten met betrekking tot de documentatie en de daarin beschreven software berusten bij Eagle Vision Systems B.V. Dit geldt ook voor eventuele aanvullingen of wijzigingen. Het gebruik van de in deze documentatie beschreven software is gebonden aan regels die worden omschreven in de licentiebepalingen. Reproductie van het materiaal, op welke wijze dan ook, is zonder voorafgaande schriftelijke toestemming uitdrukkelijk verboden. De informatie in deze documentatie kan zonder voorafgaande mededeling worden gewijzigd en impliceert geen enkele verplichting voor Eagle Vision.

# 1. APPLICATION OVERVIEW

## 1.1 Introduction

Eagle Area Monitor is a software system for video surveillance with multiple cameras. The system detects area events like person crossing a line or entering a zone of people visible in the tracking area. The detected events are provided online to 3<sup>rd</sup> party applications via a network interface.

Eagle Area Monitor employs multiple stereo cameras (EagleEyes) for (pseudo) 3-dimensional measurements of trajectories. A person measurement includes location within a two-dimensional ground plane, and a 3-dimensional bounding box (a cuboid) approximating person's volume. A time-stamped series of such measurements forms a trajectory. The EagleTrackingEngine makes **tracks** from the detected people and sends those to the Eagle Area Monitor. This application doesn't focus on a track (person) but on an area.

Trajectory measurements are streamed online to 3-rd party client applications via an Ethernet interface. The stream of measurements is organized into messages. Each message corresponds to one area event, such as: person entered a place, person left a place, person crossed a line.

Each message is sent as a separate according to the UDP/IP transport protocol.

The Eagle Area Monitor can supervise as much areas as needed, each of it having a unique ID (area ID).

## 2. APPLICATION INTERFACE

### 2.1 Physical Interface

#### 2.1.1 Physical layer

|                  |  |
|------------------|--|
| Connection type: | Ethernet   |
| Physical link:   | Cat 5 Ethernet cable with two RJ45 connectors or wireless Ethernet connection IEEE 802.11b/g standard. |
| Bandwidth:       | 10 Mb/s (megabit per second)   |
| Transport layer: | UDP / IP   |

#### 2.1.2 Transport layer

|                       |                      |
|-----------------------|----------------------|
| Protocol:             | UDP / IP             |
| Mode                  | Broadcast or unicast |
| IP broadcast port     | 52222                |
| IP subnetwork         | to be defined        |
| SPI Scout IP address: | to be defined        |

#### 2.1.3 Application layer

|              |   |
|--------------|---|
| Protocol:    | Custom SPI-AM protocol using UDP packets                        |
| Granularity  | single UDP packet per event                                     |
| Message size | variable  |
| Formatting   | JSON encoded structure,<br>formatted as text with UTF8 encoding |
| Frequency:   | variable, approx 10 messages/area/second                        |

## 2.2 General message formatting

The SPI Comm can be configured to encode tracking messages using either JSON or XML data exchange format. Both formats deliver the same content.

### JSON configuration

|              |  |
|--------------|--|
| JSON version | 1.0  |
| Encoding     | UTF-8  |
| Definition   | See: <a href="http://www.json.org">http://www.json.org</a> |

### Number formatting

|                     |   |
|---------------------|---|
| Decimal separator   | dot   |
| Thousands separator | not used  |
| Notation            | ordinary decimal notation   |
| Example             | 10230.45<br>(ten thousands two hundred thirty and fort five hundreds) |

### Time formatting

Time will be formatted using the following encoding (derived from ISO 8601 standard)

**Syntax**    yyyyymmddThhuuss.rrrrrrr

|          |  |
|----------|--|
| yyyy     | 4 digits, year                           |
| mm       | 2 digits, month within the year          |
| dd       | 2 digits, day within the month           |
| <b>T</b> | fixed literal "T"                        |
| hh       | 2 digits, hour within the day            |
| uu       | 2 digits, minute within the hour         |
| ss       | 2 digits, second within the minute       |
| rrrrrrrr | 9 digits, microseconds within the second |

**Example**    20101004T121931.556875

represents 4 October 2010, UTC time 12:19:31 and 556875 microseconds.

Time reference will be UTC (zero offset time zone). Consequently, time zone suffix will be omitted from formatted time representation.

## 2.3 JSON Message definition

A message is defined as a JSON structure with the following fields – name/value pairs:

| Field Name        | Value Type            | Description  |
|-------------------|-----------------------|--|
| format            | string                | Identifies message format  |
| version           | unsigned int (32 bit) | Identifies message format  |
|                   |                       |  |
| area id           | unsigned int (32 bit) | Unique id of the monitored area  |
| timestamp iso utc | formatted string      | Timestamp of the event, in coordinated universal time (UTC) reference, formatted as ISO time string. |
| event id          | unsigned int (32 bit) | Identifies the type of message: Person entered, Person left, PersonCountChanged...                   |
| event type        | string                | Identifies the type of message: Person entered, Person left, PersonCountChanged...                   |

Depending of the EventType, different other parameters can follow. Most of the events have a counter. The meaning of the counter is dependent of the Event Type. See the Event Description.

The following sections provide detailed definition of JSON message fields.

#### 2.3.1.1 Format

|                 |   |
|-----------------|---|
| JSON field name | "format"                                  |
| Value range     | fixed value "evs-am"                      |
| Type            | string                                    |
| Description     | Indicates communication protocol version. |

#### 2.3.1.2 Version

|                 |   |
|-----------------|---|
| JSON field name | "version"   |
| Value range     | fixed value: 2                                      |
| Type            | numeric, unsigned integer, 32 bit                   |
| Description     | Indicates version of communication protocol version |

#### 2.3.1.3 Area Id

|                 |                                   |
|-----------------|-----------------------------------|
| JSON field name | "area id"                         |
| Value range     | 0-4294967296                      |
| Type            | numeric, unsigned integer, 32 bit |
| Description     | Area identifier (unique).         |

### 2.3.1.4 Timestamp

|                 |  |
|-----------------|--|
| JSON field name | "timestamp iso utc"                      |
| Value range     | "yyyymmddThhmss.uuuuuu"                  |
| Type            | string                                   |
| Description     | Timestamp of the event in UTC reference. |

### 2.3.1.5 Event Id

|                 |                                   |
|-----------------|-----------------------------------|
| JSON field name | "event id"                        |
| Value range     | 0-4294967296                      |
| Type            | numeric, unsigned integer, 32 bit |
| Description     | Event identifier.                 |

### 2.3.1.6 Event Type

|                 |   |
|-----------------|---|
| JSON field name | "event type"  |
| Type            | string  |
| Description     | Event description. This makes double use with the Event Identifier, but can be used for display |

The Event Identifier maps the possible detected events. Each event has extra parameters that are described further. The counters have different meaning for each event, see the Event Description. Some of the events are triggered by the events themselves (marked E), others are triggered by a timer(T). The "Person Entered" event for example will come every time that a person has entered the area. The "Person Count Changed" Event will come on timer, if the number of people has changed between this and the previous timer. This message is send only once if several people enter or leave the room in the timer delay. If one person enters and 1 leaves at the same time, this message is not send. The timer update time is a setting, default 100ms. The Timer messages send the current counter value and eventually the old counter value. They can't give information about the track(s). The Event triggered messages are send out with the current counter and a track information. If the same event happens at the same time for 2 or more tracks, there will be 1 message per track.

In the following table, **A**: detection applies to areas, **L** : detection applies to lines , **T** are Timer triggered events, **E** are people triggered events. The possible events are:

| <b>Id</b> | <b>Event Name</b>      | <b>Event Description</b>                          | <b>Additional Parameters</b> |
|-----------|------------------------|---|------------------------------|
| <b>0</b>  | Presence Detected (AT) | Notifies presence , only by frist person send out |                              |

|                              |                                       |  |  |
|------------------------------|---------------------------------------|--|--|
| <b>1</b>                     | Absence Detected(AT)                  | Notifies (local) absence detection   |  |
| <b>2</b>                     | Person Count Changed(AT)              | Notifies a change of the number of tracks in an area   | - previous number of tracks<br>- current number of tracks  |
| <b>3</b>                     | Person Entered(AE)                    | Notifies that a person has entered an area. <b>Counts</b> the number of people that <b>have entered the area</b> since last reset.   | - current track counter<br>- tracks ID<br>- Position XY<br>- Height<br>- Bounding box  |
| <b>4</b>                     | Person Exit(AE)                       | Notifies that a person has left an area . <b>Counts</b> the number of people that <b>have left the area</b> since last reset.  | - current track counter<br>- tracks ID<br>- Position XY<br>- Height<br>- Bounding box  |
| <b>5</b>                     | Line Cross Count Changed(LE)          | Notifies that a person has crossed a line. Counts the <b>number of times that this line was crossed.</b>   | - current track counter<br>- track ID<br>- Position XY<br>- Height<br>- Bounding box   |
| <b>6</b>                     | Line In Count Changed(LE)             | Notifies that a person has crossed a line in the in direction. Counts the number of tracks that have crossed the line <b>in this direction</b> since the reset.  | - current track counter<br>- track ID<br>- Position XY<br>- Height<br>- Bounding box   |
| <b>7</b>                     | Line Out Count Changed(LE)            | Notifies that a person has crossed a line in the out direction. Counts the number of tracks that have crossed the line <b>in this direction</b> since the reset.   | - current track counter<br>- track ID<br>- Position XY<br>- Height<br>- Bounding box   |
| <b>8</b>                     | Person Line Crossed Count Changed(LT) | Notifies that the <b>number of persons</b> crossing the line has changed. <b>A person is counted only ones</b> if it crosses the line several times. If the person leaves the detection area, it will be considered again. | - current number of tracks   |
| <b>9 (in a later state)</b>  | MovementDetected                      | Notifies detection of movement   | - track ID<br>- Position XY<br>- Height<br>- Bounding box  |
| <b>10 (in a later state)</b> | ActivityDetected                      | Notifies detection of person activity  | -ID of person(s) moving<br>- type of activity {sitting, standing, walking, ...}<br>- Position XY<br>- Height<br>- Bounding box |

|    |                    |  |  |
|----|--------------------|--|--|
| 11 | Person Entered(AE) | Notifies that a person has entered an area. <b>Counts</b> the number of people that <b>have entered the area</b> since last reset. | <ul style="list-style-type: none"> <li>- tracks ID</li> <li>- Position XY</li> <li>- Height</li> <li>- Bounding box</li> <li>- Visit Duration</li> </ul>       |
| 12 | Person Exit(AE)    | Notifies that a person has left an area . <b>Counts</b> the number of people <b>that have left the area</b> since last reset.      | <ul style="list-style-type: none"> <li>- tracks ID</li> <li>- Position XY</li> <li>- Height</li> <li>- Bounding box</li> </ul>                                 |
|    | Line Crossed (LE)  | Notifies that a person has crossed a line.   | <ul style="list-style-type: none"> <li>- current track counter</li> <li>- track ID</li> <li>- Position XY</li> <li>- Height</li> <li>- Bounding box</li> </ul> |

### 2.3.1.7 Current number of tracks

|                 |                                   |
|-----------------|-----------------------------------|
| JSON field name | "track count"                     |
| Value range     | 0-4294967296                      |
| Type            | numeric, unsigned integer, 32 bit |
| Description     | Event identifier.                 |

### 2.3.1.8 Track Id

Each track (representing a tracked object/person over the time) has a unique Id. This remains the same as long as the camera follows the person/object.

|                 |                                   |
|-----------------|-----------------------------------|
| JSON field name | "track id"                        |
| Value range     | 0-4294967296                      |
| Type            | numeric, unsigned integer, 32 bit |
| Description     | Person identifier.                |

### 2.3.1.9 Position XY

|                 |  |
|-----------------|--|
| JSON field name | "position xy"  |
| Value range     | approx. (-10.0...10.0) - depending on tracking area  |
| Type            | array of two floating-point numbers  |
| Description     | The (x,y) coordinates of track position, relative to system coordinate system. <b>Unit:</b> meter. |

### 2.3.1.10 Height

|                 |  |
|-----------------|--|
| JSON field name | "height"   |
| Value range     | approx. (0...3.0) - depending on tracked track                 |
| Type            | floating point number  |
| Description     | Height of the tracked track, non-negative. <b>Unit:</b> meter. |

### 2.3.1.11 Bounding box

|                 |  |
|-----------------|--|
| JSON field name | "bbox xyzxyz"  |
| Value range     | approx. (-10.0...10.0) - depending on tracking area  |
| Type            | array of six floating-point numbers  |
| Description     | This field indicates a 3-dimensional bounding box of the tracked track. The bounding box is given as two 3D dimensional points, in the following order:<br>x_min            x-coordinate of box minimum corner<br>y_min            y-coordinate of box minimum corner<br>z_min            z-coordinate of box minimum corner<br>x_max            x-coordinate of box maximum corner<br>y_max            y-coordinate of box maximum corner<br>z_max            z-coordinate of box maximum corner<br><b>Unit:</b> meter. |

### 2.3.1.12 Visit Duration

|                 |  |
|-----------------|--|
| JSON field name | "visit duration"                                     |
| Value range     | "yyyymmddThhmmss.uuuuuu"                             |
| Type            | string   |
| Description     | Visit duration, send out when a person exit an area. |

### 2.3.1.13 Example

An example message formatted as a JSON structure:

```
{
  "format": "evs-am",
  "version": 2,
  "area id": 6,
  "timestamp iso utc": "20120327T140532.686014",
  "track count": 4,
  "track id": 4,
  "visit duration": 0,
  "position ": [0, 0],
  "height": 0,
  "bbox xyzxyz": [0, 0, 0, 0, 0, 0],
  "event id": 5,
  "event type": "Line Cross Count Changed"
```

```
}
```

Notice that the format specification does not define the order of the fields in the structure. The client/receiver application should not assume any specific order of the fields in the message.

## 2.4 Reference coordinate system

## 2.5 Temporal reference

All timestamps are expressed in UTC reference (Universal Coordinate Time), which is independent of local time zone and daylight settings.

The SPI Comm applications can be time-synchronized with external systems using NTP network protocol.

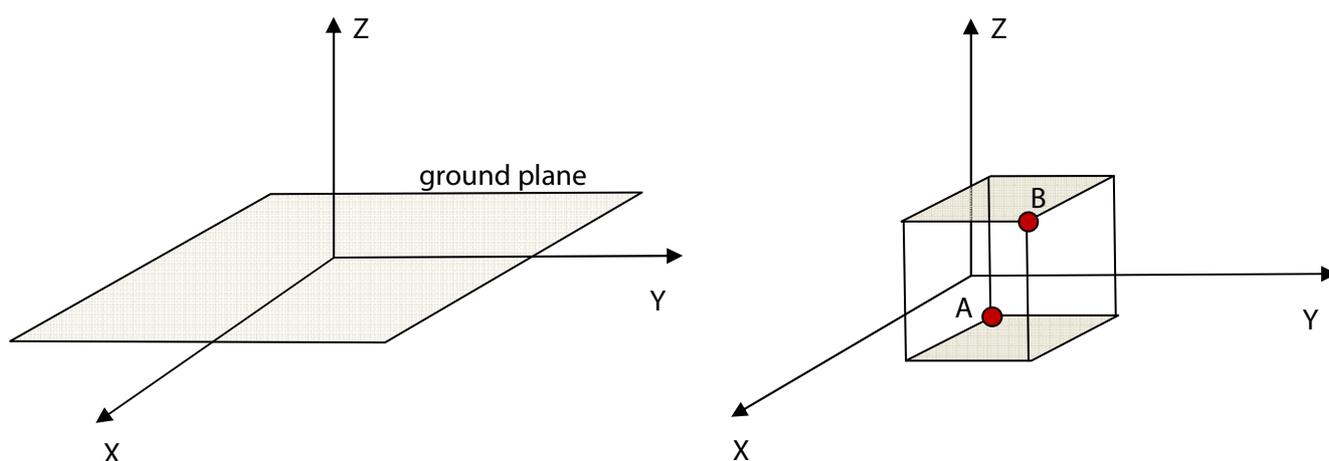
## 2.6 Spatial coordinate system

The SPI Scout system is using a 3-dimensional, right-handed, Cartesian coordinate system. All coordinates are expressed in meters. The X and Y axes define the ground plane of the coordinate system, the Z axis is the vertical axis.

The coordinate system is chosen such that the center point (0,0,0) is visible at one of the cameras. The value range for the X and Y coordinates depends on the size of the visible area. In a typical configuration the X and Y coordinates fall into range from -10.0 to +10.0 meter. The Z-plane is chosen such that Z=0 corresponds to the ground floor. The value range for the Z coordinates is 0.0 to 3.0 meters.

### 2.6.1 Bounding box measurement

Bounding box is a 3-dimensional cuboid, defined by two characteristic points: (a) the minimal point and (b) the maximal point – as indicated in the Figure 1. The height of the bounding box corresponds to the height of the person.



**Figure** (Left) Coordinate system and ground plane.

(Right) Bounding box definition, A – minimum point, B – maximum point.

