



FIGHTER UAV COMPETITION
COMMUNICATION DOCUMENT

CONTENT

1	Purpose	1
2	Connection	1
3	Status Codes	2
4	API Addresses	2
5	Login to the Server	3
6	Sign-Out	3
7	Server Time	3
8	Telemetry	4
8.1	Telemetry Packet	4
8.2	Sample Telemetry Data.....	5
8.3	Sample Telemetry Answer	6
9	Sending Locking Information.....	7
9.1	Sending Locking Data	7
10	Submission of Kamikaze Data	8
10.1	Sample Kamikaze Data	8
10.2	Sample QR Code	8
11	Obtaining QR Coordinates	9
11.1	Sample QR Coordinate Data	9

FIGURES

Figure 1 All network diagram	1
Figure 2: Sample QR code	8

1. Purpose

This document contains information on how to ensure communication between the host of the Fighter UAV competition to be held within the scope of Teknofest 2022 and the teams.

2. Connection

Teams must communicate with the contest server during the competition to send telemetry and crash information, and to obtain the system time and location information of other aircraft.

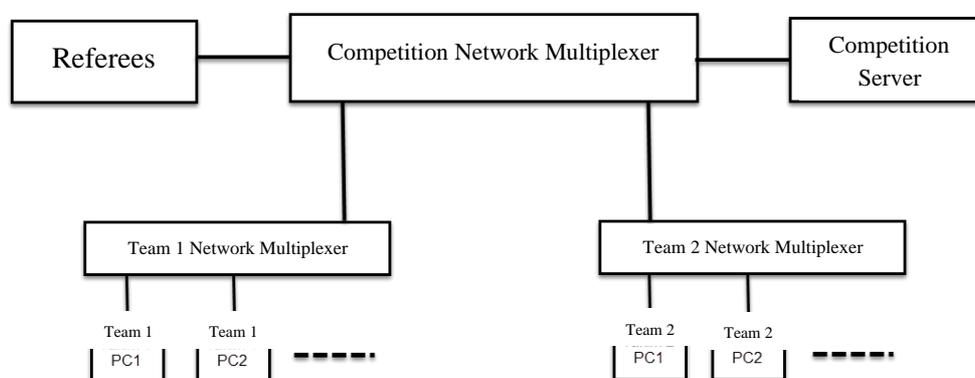


Figure 1 All Network System

During the competition, the teams will be provided with an ethernet cable so that they can connect to the local network in which the competition server is located. Each team must be connected to the competition network via this ethernet cable with only one ip address. During the competition, an ip address will be specified to the teams and only connection to the system via the specified ip addresses will be allowed. Teams will be able to open the competition network in their local network through this ip address.

Contest host; The actual address will be determined on the day of the competition and will have an address in `http://127.0.0.25:5000` format. All communications with the server will be in JSON format with API logic.

3. Status Codes

The following HTTP status codes may be obtained as a result of the communication made with the competition server via the API.

- **200:** Request succeeded
- **204:** Format of sent packet Incorrect
- **400:** Request is incorrect or invalid. In such a case, the error code is sent as the page content.
Descriptions of the error codes are included in the Error codes heading.
- **401:** An attempt to access identityless. You need to sign in.
- **403:** Unauthorized access attempt. An account that does not have administrator privileges is trying to log in to private links for administrators.
- **404:** Invalid URL.
- **500:** Intra-server error.

4. API Addresses

- **GET /api/servertime:** Used to query the server time. Detailed information about the server time and the format of the server time are described in the *Server Time* heading.
- **POST /api/telemetri_gonder :** Used to instantly send the aircraft's information to the server and to receive the information of other teams. Detailed information is provided in the *Telemetry* topic.
- **POST /api/kilitlenme_bilgisi :** After a successful crash on a competitor UAV, crash information is sent over this connection. *Detailed information is given in the heading Sending Crash Information.* x **POST /api/login :** Used for sign-in using the team-specific username and password. How to log in to the system is described in the Login to Server topic.
- **GET /api/exit :** The contest session is closed at this address. Teams that have completed a round of competition must close their session with this address. Detailed information is provided in *Logout*.
- **POST /api/kamikaze_bilgisi:** The text read after the Kamikaze task is sent over this link. *Detailed information is given in the heading of sending Kamikaze Information.*
- **GET /api/qr_koordinati:** Sends back the position of the QR code that will be used for the Kamikaze task in the competition area. Detailed information is given under the heading *Obtaining QR Coordinates*.

5. Login to the Server

During the competition, teams must first log in to receive information from the server and send information. The login process must be done once before the start of the contest. If the connection is lost, you can log in again. Teams will be given their username and password information before the competition, with which they can log in to the system. Queries made without signing in are answered with the status code "401 Anonymous Access Attempt".

In order to log in, the information given below should be posted to `/api/login`.

```
{
  "kadi" : "takimkadi",
  "sifre" : "takimsifresi"
}
```

If the entry is successful, the team number is taken as the content along with the 200 OK status code. If the username or password is invalid, a 400 status code is received as the answer.

6. Sign-out

Teams that complete the competition session must close their sessions to stop the competition period. Teams that do not log out are considered in the competition and may receive penalties such as a timeout penalty or a penalty for low telemetry sending speed. For the sign-out process, it is sufficient to go to `/api/cikis` once with the Get method. Teams will have their sessions terminated by the referees due to accident-breakdown or when they land in fail-safe mode. Teams that end their session once in a competition round are considered to have completed that round and cannot re-enter.

7. Server Time

The server time must be added to the packages sent so that all data is synchronized during the competition. The server time can be queried from the `/api/servertime` connection, and teams are advised to calibrate the clocks of their computers to communicate with the server with the specified server time.

All server times received and sent are in the following format:

```
{
  "current": 6,
  "minutes": 9,
  "seconds": 2,
  "milliseconds": 617
}
```

8. Telemetry

Teams must send data indicating the status of the UAV to the server at least 1 Hz per second, as specified in the competition specification. Telemetry packets sent over 2 Hz are answered with 3 error codes as page content along with *a status code of 400*. The data that should be included in the telemetry packet and its descriptions are described in the Telemetry Packet Data topic. The JSON data specified in the *Sample Telemetry Data* header should be posted to the *api/telemetri_gonder* http://127.0.0.25:5000/api/telemetri_gonder address. In response to this post, teams will be able to receive the location information of the other competitors along with the presentation time. Within this location information, the time difference between the location information and the server time will be given in milliseconds.

8.1 Telemetry Packet Data

- **takim_numarasi:** The team number given to the team by the referees.
- **IHA_enlem:** The aircraft's latitude information in decimal form.
- **IHA_boylam:** The longitude information of the aircraft in decimal form.
- **IHA_irtifa:** The height of the aircraft in meters relative to the ground.
- **IHA_dikilme:** The angle of rotation of the aircraft in degrees.
- **IHA_yonelme:** Orientation of the aircraft to the north in degrees
- **IHA_yatis:** Angle of lean of the aircraft in degrees
- **IHA_hiz:** The ground speed of the aircraft in meters/second.
- **IHA_batarya:** The IHA_otonom of the aircraft's battery or fuel in percentage terms: The information whether the aircraft is in autonomous flight mode. This value; it should be 1 if the aircraft is autonomous and 0 if not.

- **IHA_kilitlenme:** Information whether there was a crash at the time the telemetry was sent. If the aircraft is trying to follow another aircraft, this value should be 1 and the following data about the target should be different from zero.
- **Hedef_merkez_X:** The horizontal component of the position of the target the aircraft is trying to follow in the image. The upper-left point of the image is considered 0, and this value increases to the right.
- **Hedef_merkez_Y:** The vertical component of the position of the target the aircraft is trying to follow in the image. The upper-left point of the image is considered 0, and this value increases downwards.
- **Hedef_genislik:** The width of the target area in the image.
- **Hedef_yukseklk:** The height of the target area in the image.
- **GPSSaati:** Clock data (UTC) of GPS data received from the aircraft. This data should be as it appears in the following example. GPSSaati must have time information coming directly from the aircraft.

8.2 Sample Telemetry Data

```
{
  "takim_numarasi": 1,
  "IHA_enlem": 43.576546,
  "IHA_boylam": 22.385421,
  "IHA_irtifa": 100,
  "IHA_dikilme": 5,
  "IHA_yonelme": 256,
  "IHA_yatis": 0,
  "IHA_hiz": 223,
  "IHA_batarya": 20,
  "IHA_otonom": 0,
  "IHA_kilitlenme": 1,
  "Hedef_merkez_X": 315,
  "Hedef_merkez_Y": 220,
  "Hedef_genislik": 12,
  "Hedef_yukseklk": 46,
  "GPSSaati": {
    "saat": 19,
    "dakika": 1,
    "saniye": 23,
    "milisaniye": 507
  }
}
```

8.3 Sample Telemetry Answer

```
{
  "sunucuSaati": {
    "saat": 6,
    "dakika": 53,
    "saniye": 42,
    "milisaniye": 500
  },
  "konumBilgileri": [
    {
      "takim_numarasi": 1,
      "iha_enlem": 12.236945,
      "iha_boylam": 26.945331,
      "iha_irtifa": 25,
      "iha_dikilme": 5,
      "iha_yonelme": 256,
      "iha_yatis": 0,
      "zaman_farki": 93
    },
    {
      "takim_numarasi": 2,
      "iha_enlem": 41.265854,
      "iha_boylam": 25.697435,
      "iha_irtifa": 55,
      "iha_dikilme": 5,
      "iha_yonelme": 256,
      "iha_yatis": 0,
      "zaman_farki": 74
    },
    {
      "takim_numarasi": 3,
      "iha_enlem": 41.598546,
      "iha_boylam": 26.974315,
      "iha_irtifa": 65,
      "iha_dikilme": 5,
      "iha_yonelme": 12,
      "iha_yatis": 4,
      "zaman_farki": 43
    }
  ]
}
```

9. Sending Locking Information

Teams must send crash information to the server after each successful locking they perform. Locking information should include when the lockdown began, when the lockdown ended, and whether the lockdown was autonomous. The time information must be of the server time type. If the locking is autonomous, the `otonom_kilitlenme` data must be 1. Locks made without sending the locking information are not subject to scoring. Locking information must be sent after the end of the lock, and only one packet must be sent for each lockdown. Submitted locking information will be scored for review using videos recorded at the end of the contest session and replay systems. An example of locking data is shown in [the Sample Locking Data](#) topic.

9.1 Sample Locking Data

```
{
  "kilitlenmeBaslangicZamani": {
    "saat": 19,
    "dakika": 1,
    "saniye": 23,
    "milisaniye": 507
  },
  "kilitlenmeBitisZamani": {
    "saat": 19,
    "dakika": 1,
    "saniye": 45,
    "milisaniye": 236
  },
  "otonom_kilitlenme": 0
}
```

10. Submission of Kamikaze Information

Teams must send kamikaze information to the server after a successful kamikaze mission. Kamikaze knowledge should include kamikaze start time, kamikaze start time and text information obtained as a result of reading the QR code. The time information must be of the server time type. Kamikaze information should be sent after the kamikaze expiration, and only one package should be sent for each kamikaze. The kamikaze information submitted will be reviewed using videos

recorded at the end of the contest session and replay systems. The example of kamikaze data is shown in *the Sample Kamikaze Data* heading. A QR code example is shown in *the Sample QR Code* heading. The QR code to be used in the competition will be Version 1.

10.1 Sample Kamikaze Data

```
{
  "kamikazeBaslangicZamani": {
    "saat": 19,
    "dakika": 1,
    "saniye": 23,
    "milisaniye": 507
  },
  "kamikazeBitisZamani": {
    "saat": 19,
    "dakika": 1,
    "saniye": 28,
    "milisaniye": 236
  },
  "qrMetni ": "teknofest2022"
}
```

10.2 Sample QR Code

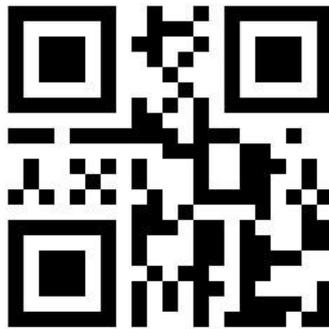


Figure 2: Sample QR Code

11. Obtaining QR Coordinates

Teams can get the location of the QR code to be used in the competition with the query they will send to the server. This information includes the latitude and longitude information of the location of the QR code. An example of a QR coordinate is given under the heading *Sample QR coordinate Data*.

11.1 Sample QR Coordinate Data

```
{  
  "qrEnlem": 41.123456,  
  "qrBoylam": 26.654987  
}
```