

# BOOTLOADER MANUAL DOCUMENT



## Index table

1- Before Starting .....	3
2- File Structure. ....	3
3- User Software Language.....	4
4- Customizing the Program .....	4
4.1- For Windows .....	4
4.2- For Linux.....	4
4.3- For Mac .....	4
5- The use of json file.....	5
5.1- buttons.json .....	5
5.1.1- id data.....	6
5.1.2- prop data .....	6
5.1.3- posi data .....	6
5.1.4- command data .....	7
5.2 - images.json .....	7
5.2.1- id data.....	8
5.2.2- prop data .....	8
5.2.3- posi data .....	8
5.2.4- imgp data.....	8
5.3- lables.json.....	9
5.3.1- id data.....	10
5.3.2- prop data .....	10
5.3.3- posi data .....	10
5.4- textbox.json.....	10
5.4.1- id data.....	11
5.4.2- prop data .....	12
5.4.3- posi data .....	12
5.4.4- multiline data .....	12
5.4.5- text data .....	12

## 1- Before Starting

This document was written by pickat on 31.03.2018. This document may have temporary information. You can visit <http://pickat.org/> to check for updated information.

## 2- File Structure.

- Bootloader
  - main.py
    - The center where all libraries are combined
  - functions.py
    - Button click event file with functions.
  - drawing.py
    - The file from which the form is drawn.
  - Assets folder.
    - Index for this folder.
      - buttons.json
        - Includes data of buttons to be drawn to the form.
      - favicon.ico
        - This is a icon file for this program.
      - images.json
        - The form contains the data of the images to be plotted.
      - labels.json
        - The form contains the data of the scripts to be plotted.
      - textbox.json
        - The form contains the TextBox data to be plotted.
      - Logo\_Black.png
        - One of the images drawn by the program we added by default.
      - mikroe-uhb.exe
        - An exe file that allows us to load the program into PickKat.
      - settings.ini
        - Contains the location of Json files.
        - Format icon, width, height, name, and whether horizontal or vertical size can be changed.
      -

The Backup and Circle directories will be used with the default plugins in the Lib directory.

## 3- User Software Language

Python Version 3.6.2

## 4- Customizing the Program

### 4.1- For Windows

You do not need a Python software to use the program. Main.exe \*.py compiles the files again every time.

It is possible to customize the program either software or design.

To customize the software is enough to have just python3 information.

You can use Python's main software is installed when the module is limited. You can use basic modules defined exception into all modules.

Modules can use this exception -Basic modules are as follows.

- 1) Pillow Module (was included as PIL)
- 2) JSON Module (were included as JSON)
- 3) ConfigParser Module (was included as ConfigParser)

To customize the program as you can by applying conceptual json file into the shape of an object available data to each object.

### 4.2- For Linux

To use the program must be installed on your computer python3. It is not offered in any package.

In order to use the program, you need to download the following modules from the python3 server.

- 1) Pillow Module.
- 2) Json Module.
- 3) Configparser Module.

To customize the software is enough to have just python3 information.

To customize the program as you can by applying conceptual json file into the shape of an object available data to each object.

### 4.3- For Mac

In order to download packages from Mac, you must first obtain the pip command for python3.

After Piper command is available, you must obtain the ActiveTcl-8.6.7.0-macosx10.9-x86\_64-404783.pkg with Tkinter module package.

You can obtain the remaining modules pips command.

- 1) Pillow Module.
- 2) Json Module.
- 3) Configparser Module.

After all modules are downloaded you may use the program.

To customize the software is enough to have just python3 information.

To customize the program as you can by applying conceptual json file into the shape of an object available data to each object.

## 5- The use of json file

### 5.1- buttons.json

Please visit [https://www.tutorialspoint.com/python/tk\\_button.htm](https://www.tutorialspoint.com/python/tk_button.htm) place to see a list of settings of the button structure.

buttons.json generic design of the file:

```
[
    { 1.buton
        The data required for drawing the button.
    },
    { 2.buton
        The data required to draw a new button more.
    }
]
```

In json format of this file is coded Please note that registration with the ANSI.

Adding buttons as you want in the form python3 stateless person above has been made easier thanks to the overall design.

There are certainly a few data that should be between the curly brackets.

Program on "**HEX Dosyası Seç**" Let us examine the data of the green button.

Data is different spellings according to the type of value they receive.

Spelling forms according to the type of data.

- 1) Exact numbers are written without quotation marks inside the no.
- 2) Decimal numbers are written with dividing point.
- 3) Text in double quotes (") between written -" This is a text "for instance.

```
{
    "id": "button1",
    "prop": {
        "text": "HEX Dosyası Seç",
        "width": 30,
        "height": 2
    },
    "posi": {
        "x": 50,
        "y": 200
    },
    "command": "button1Click"
},
```

#### 5.1.1- id data

Above background with yellow "id" data should definitely be on every button.

This data is used for the separation of programs running object. So there should be no other object with the same id that the id of this data.

Writing must be formed using only letters and numbers.

When used as selective in the program should be considered case-compliance.

#### 5.1.2- prop data

It is not very important whether or not there is data in the section. However, even if there is no data, it is necessary to find it.

In the buttons, the default value of the "text" data in the prop is set to "not defined".

You can learn the names and values of the data found here by visiting the address below.

[https://www.tutorialspoint.com/python/tk\\_button.htm](https://www.tutorialspoint.com/python/tk_button.htm)

#### 5.1.3- posi data

This data contains 2 different data. The names of the data in it are fixed and both must be exist. Also it is mandatory that each POSI data of the data button.

Wherein x and y are two data including. This is an integer from two data values.

X data button in a horizontal as "pixel" is set where it would be.

Y enter the data in the vertical as "pixel" is set where it would be.

### 5.1.4- command data

This data set button click event. The text you write here should consist of only letters and numbers. They should not contain Turkish characters. Also, the first character in this field can not be a number.

The value you specify as the command must be written as a function in the functions.py file.

This function takes "this" as a parameter. Within this value, the button has the "id" data.

Please refer to the section titled “**Programı Özelleştirmek**”, which is the 4th place in the function.

### 5.2 - .json görüntüleri

Art is the same as infrastructural label.

To see a list of settings of the tag structure [https://www.tutorialspoint.com/python/tk\\_label.htm](https://www.tutorialspoint.com/python/tk_label.htm) please visit the address.

images.json overall design of the file:

```
[
    { 1.resim
        The data required to draw the picture.
    },
    { 2.resim
        The data required to draw a new picture.
    }
]
```

In json format of this file is coded Please note that registration with the ANSI.

Thanks to the design form in general above to add pictures as you want python3 stateless person has been made easier.

There are certainly a few data that should be between the curly brackets.

Let us examine the data pickat logo on the program.

Data is different spellings according to the type of value they receive.

Spelling forms according to the type of data.

- 1) Exact numbers are written without quotation marks inside the no.
- 2) Decimal numbers are written with dividing point.
- 3) Text in double quotes (") between written -" This is a text "for instance.

```

{
    "id": "image1",
    "prop": {
        "width": 580,
        "height": 167
    },
    "posi": {
        "x": 10,
        "y": 10
    },
    "imgp": {
        "src": "assets/Logo_Black.png",
        "width": 580,
        "height": 167
    }
}

```

### 5.2.1- id data

Above background with yellow "id" data should definitely be on every button.

This data is used for the separation of programs running object. So there should be no other object with the same id that the id of this data.

Writing must be formed using only letters and numbers.

When used as selective in the program should be considered case-compliance.

### 5.2.2- prop data

It is not very important whether or not there is data in the section. However, even if there is no data, it is necessary to find it.

You can learn the names and values of the data found here by visiting the address below.

[https://www.tutorialspoint.com/python/tk\\_label.htm](https://www.tutorialspoint.com/python/tk_label.htm)

### 5.2.3- posi data

This data contains 2 different data. The names of the data in it are fixed and both must be exist. Also POSI is obligatory on every image data of the data.

Wherein x and y are two data including. This is an integer from two data values.

X data in the image horizontally as "pixel" is set where it would be.

Y data in the image horizontally as "pixel" is set where it would be.

### 5.2.4- imgp data

This data takes a minimum of 1 up to 3 different values. Each image must reside in the object.

This data within the "src" indicates the file location data as text. There are two rules when writing this file location: "File system naming and index delimiters may be different."

- 1) If the picture main.py "could be compiled as a package main.exe." If a directory under the directory where the file directory as the first character can not be put between brackets. (It is defined as subdirectories pickat the logos as an example. In addition, fixed brackets instead of the slash between your operating system uses as the default directory "/" is used.)
- 2) If the image should be written in another directory, the directory where exactly the picture. (Eg C:\image.png "for windows" or ~/image.png "for linux and mac.")

### 5.3- lables.json

To see a list of settings of the tag structure [https://www.tutorialspoint.com/python/tk\\_label.htm](https://www.tutorialspoint.com/python/tk_label.htm) please visit the address.

lables.json overall design of the file:

```
[
    { 1. Label
        The data required to draw the label.
    },
    { 2. Label
        The data required to draw a new label.
    }
]
```

In json format of this file is coded Please note that registration with the ANSI.

Thanks to the design form in general above to add pictures as you want python3 stateless person has been made easier.

There are certainly a few data that should be between the curly brackets.

Let us examine the data pickat logo on the program.

Data is different spellings according to the type of value they receive.

Spelling forms according to the type of data.

- 1) Exact numbers are written without quotation marks inside the no.
- 2) Decimal numbers are written with dividing point.
- 3) Text in double quotes (") between written -" This is a text "for instance.

```
{
    "id": "label1",
    "prop": {
        "text": "PicKat Open Source Hid Programmer Gui"
    },
}
```

```
    "posi":{
        "x":10,
        "y":370
    }
}
```

### 5.3.1- id data

Above background with yellow "id" data should definitely be on every button.

This data is used for the separation of programs running object. So there should be no other object with the same id that the id of this data.

Writing must be formed using only letters and numbers.

When used as selective in the program should be considered case-compliance.

### 5.3.2- prop data

It is not very important whether or not there is data in the section. However, even if there is no data, it is necessary to find it.

Tags also in the prop "text" data default is "not defined" is set.

You can learn the names and values of the data found here by visiting the address below.

[https://www.tutorialspoint.com/python/tk\\_label.htm](https://www.tutorialspoint.com/python/tk_label.htm)

### 5.3.3- posi data

This data contains 2 different data. The names of the data in it are fixed and both must be exist. Also POSI is obligatory on every image data of the data.

Wherein x and y are two data including. This is an integer from two data values.

X data in the image horizontally as "pixel" is set where it would be.

Y data in the image horizontally as "pixel" is set where it would be.

## 5.4- textbox.json

textbox.json overall design of the file:

```
[
    { 1. Text box
        The data required for drawing text box.
    },
    { 2. Text box
        The data required for drawing new text box.
    }
]
```

]

In json format of this file is coded Please note that registration with the ANSI.

Thanks to the design form in general above to add pictures as you want python3 stateless person has been made easier.

There are certainly a few data that should be between the curly brackets.

Let us examine the data pickat logo on the program.

Data is different spellings according to the type of value they receive.

Spelling forms according to the type of data.

- 1) Exact numbers are written without quotation marks inside the no.
- 2) Decimal numbers are written with dividing point.
- 3) Text in double quotes (") between written -" This is a text "for instance.

```
{
  "id": "textBox1",
  "prop": {
    "width": 33,
    "height": 13,
    "bg": "black",
    "fg": "green"
  },
  "posi": {
    "x": 300,
    "y": 150
  },
  "multiline": true,
  "text": "Terminal Çıktısı"
}
```

#### 5.4.1- id data

Above background with yellow "id" data should definitely be on every button.

This data is used for the separation of programs running object. So there should be no other object with the same id that the id of this data.

Writing must be formed using only letters and numbers.

When used as selective in the program should be considered case-compliance.

#### **5.4.2- prop data**

It is not very important whether or not there is data in the section. However, even if there is no data, it is necessary to find it.

You can learn the names and values of the data found here by visiting the address below.

[https://www.tutorialspoint.com/python/tk\\_entry](https://www.tutorialspoint.com/python/tk_entry)

#### **5.4.3- posi data**

This data contains 2 different data. The names of the data in it are fixed and both must exist. Also POSI is obligatory on every image data of the data.

Wherein x and y are two data including. This is an integer from two data values.

X data in the image horizontally as "pixel" is set where it would be.

Y data in the image horizontally as "pixel" is set where it would be.

#### **5.4.4- multiline data**

The writing of this data is not mandatory. When the articles are true and get two different values to be false.

This value is not written to take false by default.

The generated text box is used to specify whether to accept multiple lines.

#### **5.4.5- text data**

Unlike the label here "text" data available in the prop.

However, with all the features as text data in the prop it has the same structure.