AIRDROITECH SDN.BHD

IR Blaster GUI Documentation

GUI Development and Features Introduction

The introduction of hardware and software setup for GUI development, and features introduction for the program

Table of Contents

Table of contents

1
Background

Objective

The establishment and development of this GUI program acts as a tool to interface with IR Blaster. This program application contains all the control commands particularly used to perform task automation to inspect the status, and performance of IR Blaster.

Program Scope

This GUI develop as a result of platform to provide user-friendly way for non-technical users, such as operator to interact with IR Blaster. This program can support the individuals that may not have command line or programming expertise to perform task and manage production processes.
This GUI tools can enforce standardized procedures and processes, ensuring that tasks are carried out consistently across different users(operators).

2
Components and Hardware Connection

Hardware Components

1. USB - TTL Converter
2. IR Blaster
3. RS485 - TTL

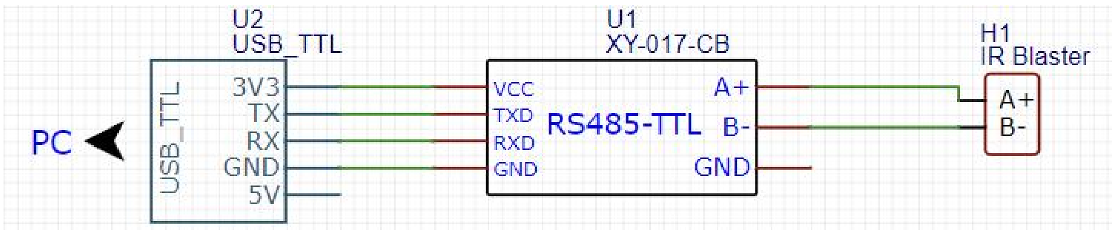
Transport Layer

RS485

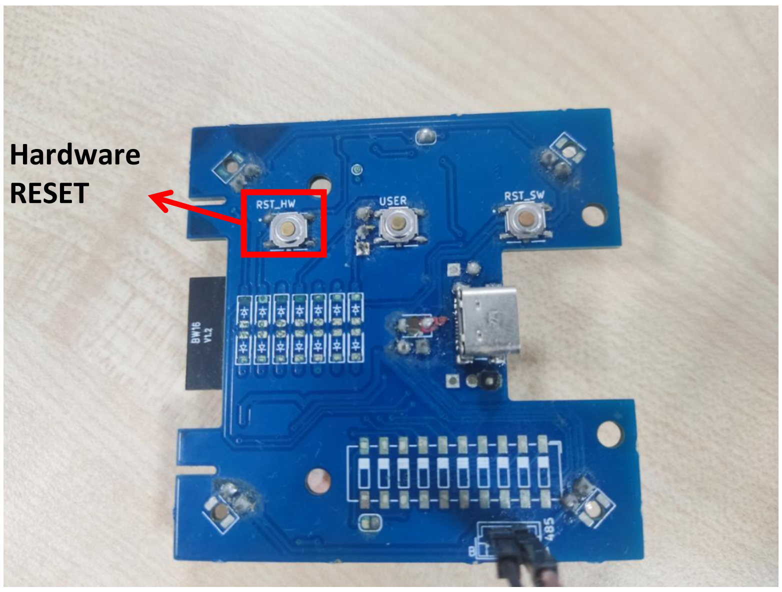
Driver

USB-SERIAL CH340

Hardware Connection



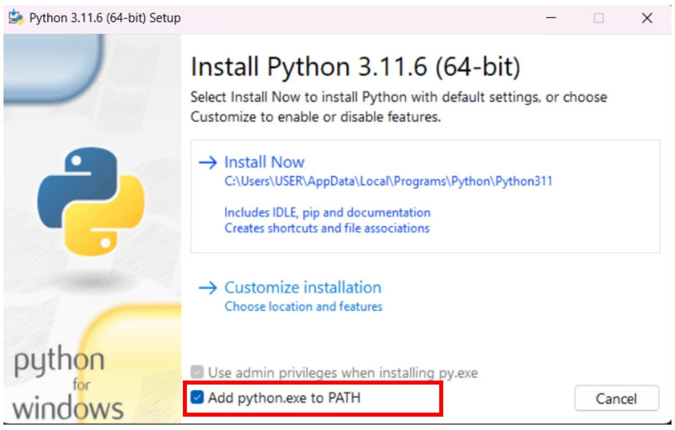
IR Blaster Board



3
Programming and Setup Environment

1. Installation of python and Tkinter

Install Python from official site python.org. The Tcl/tk packages required for GUI development (Tkinter) will be installed together.


Python 3.11.6 (64-bit) Setup

Install Python 3.11.6 (64-bit)

Select Install Now to install Python with default settings, or choose Customize to enable or disable features.

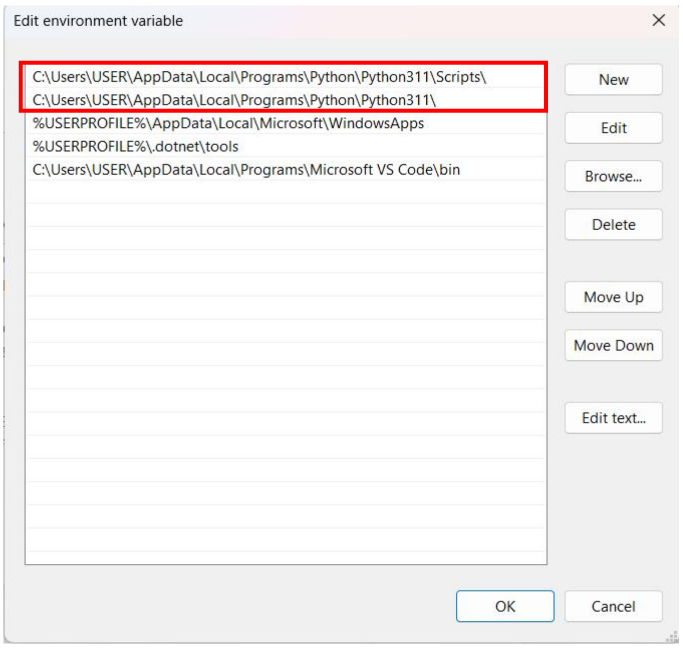
→ Install Now
C:\Users\USER\AppData\Local\Programs\Python\Python311

Includes IDLE, pip and documentation
Creates shortcuts and file associations

→ Customize installation
Choose location and features

Use admin privileges when installing py.exe
[✓ Add python.exe to PATH]

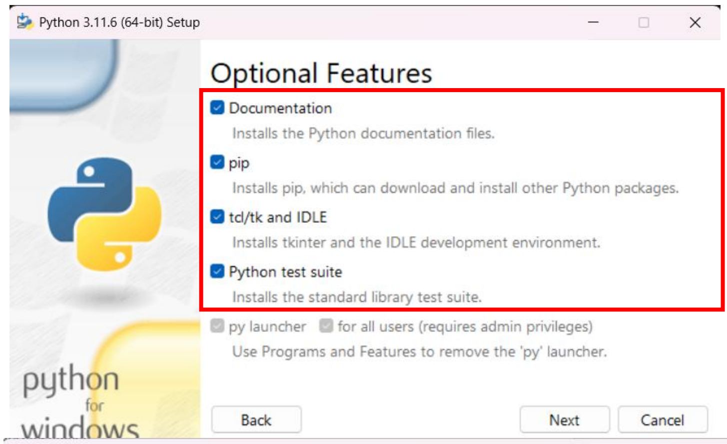
Tick the "Add python executable file to PATH" will add path to user


Edit environment variable

C:\Users\USER\AppData\Local\Programs\Python\Python311\Scripts\
C:\Users\USER\AppData\Local\Programs\Python\Python311\
%USERPROFILE%\AppData\Local\Microsoft\WindowsApps
%USERPROFILE%\.dotnet\tools
C:\Users\USER\AppData\Local\Programs\Microsoft VS Code\bin

3
Programming and Setup Environment

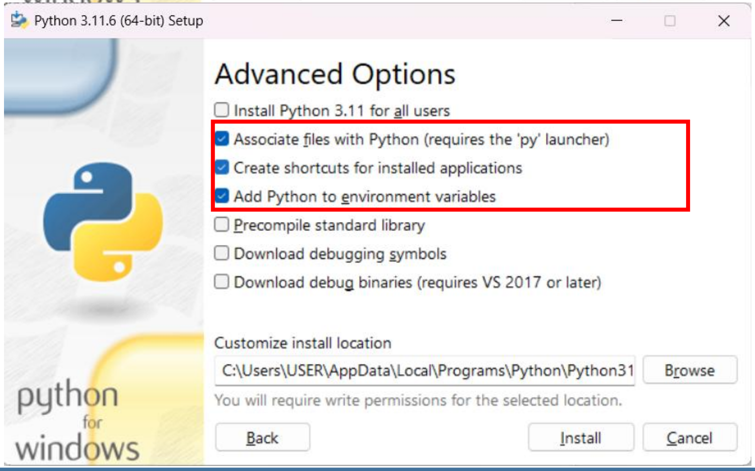
Include the features of tkinter



Python 3.11.6 (64-bit) Setup

Optional Features

- [✓] Documentation
  Installs the Python documentation files.
- [✓] pip
  Installs pip, which can download and install other Python packages.
- [✓] tcl/tk and IDLE
  Installs tkinter and the IDLE development environment.
- [✓] Python test suite
  Installs the standard library test suite.
- [✓] py launcher [✓] for all users (requires admin privileges)
  Use Programs and Features to remove the 'py' launcher.



Python 3.11.6 (64-bit) Setup

Advanced Options

- [ ] Install Python 3.11 for all users
- [✓] Associate files with Python (requires the 'py' launcher)
- [✓] Create shortcuts for installed applications
- [✓] Add Python to environment variables
- [ ] Precompile standard library
- [ ] Download debugging symbols
- [ ] Download debug binaries (requires VS 2017 or later)

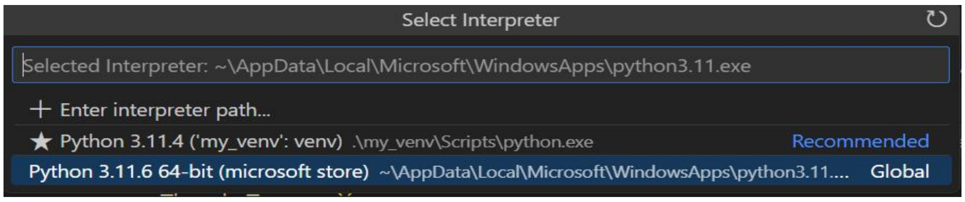
Customize install location

C:\Users\USER\AppData\Local\Programs\Python\Python31

You will require write permissions for the selected location.

3
Programming and Setup Environment

2. Setup Extension in IDE (VS Code)



Select Interpreter

Selected Interpreter: ~\AppData\Local\Microsoft\WindowsApps\python3.11.exe

+ Enter interpreter path...

★ Python 3.11.4 ('my_venv': venv) .\my_venv\Scripts\python.exe Recommended

Python 3.11.6 64-bit (microsoft store) ~\AppData\Local\Microsoft\WindowsApps\python3.11.... Global

Integrated Development Environment (IDE)

Visual Studio Code

Libraries imported and functionalities

Libraries	**Functionalities**
Tkinter	Python interface to Tcl/tk GUI toolkit
Tkinter - ttk	Themed tkinter widget set, (buttons, labels, entry field) provide enhanced version of widgets view and look
Serial (Pyserial)	Interface with serial ports
Tkinter - messagebox	Prompt indication window to user
Threading	Perform multiple tasks simultaneously
Queue	To manage task to follow “first-in, first-out”(FIFO) principle
Serial.tools.list_ports	To search available serial ports connected to device
Configparser	To read and write the config data stored in plain text file (INI format)
Json	Convert data to json format string
Os	Interact with operating system and manage file system
Base64	Particularly used for password encoding
Time	Start timer
Sys	Find the path of where python interpreter executable file is
Note: If any module library cannot be accessed, install the module library using pip	

3
Programming and Setup Environment

Code method explanation

Class	Method	Function
SerialConfigurationWindow	1. __init__	Initialize config window, preset config settings
	2 create_configuration _widgets	setting all widgets in configuration window
	3 save_settings	save preset config settings from __init and save in config file
	4. load_settings	find title in config file, if found, get the key value.
	5. set_default_settings	set the config settings to a predefined default settings.
SerialCommunicationApp	1. __init__	1. Initialize the main window 2. set attributes of windows 3. set config file for password entry 4. widget manager to monitor status of window 5. Find available ports when running GUI 6. set the frame(container for widget) in frontend window. 7. Set the flag for sending data, receiving data, and pausing data 8. Generate window for front end window and prompt window
	2 front_window	Set frontend widgets

	3 auto_prompt_window	1. Set widget for prompt window 2. Force attention on the window 3. Disable interaction with main window when present 4 Check status of prompt window
	4 find_config_file	1 Set the widget for find config file window 2 Find path of current script file and executable file and read the config file available in that directory 3 Add listbox to list the config file available
	5 apply_config_file	1 from the second children widget(listbox) of find config file window, get the selection from listbox and parse the content of the config file. 2. get all the key value pairs of config file
	6 test_print	print the key and value from section of config file loaded
	7 auto_mode	close the prompt window and open config file window to select config file for auto mode
	8 manual_mode	Set the GUI to manual mode
	9 compiled_select_config_close	apply config file, close find config file window, create port connection window
	10 create_port_connection_window	Create port connection window, if port open, the window auto close
	11 open_command_panel	1. Compile all the command into a single panel window 2. every category of

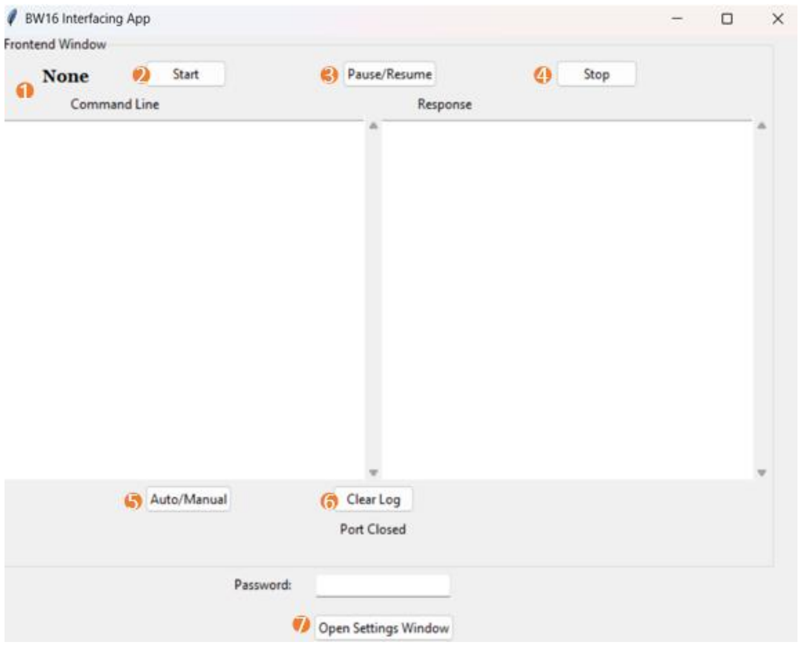
		command has the frame, if no section available in config file, add section with all the command in the series defined 3. set the label and command to config file based on label & data given in code.
	12 compiled_function	display commandline in front end display and data in backend data sending monitor
	13 frontend_display	used in method(**11**) to print the key of config file in monitor
	14 create_password	encode password and saved in config.ini
	15 send_command_data	used in method(**11**) to send command data to backend monitor
	16 update_port_status	update port status when close or open
	18 receive_data	Set the flag of receive thread, configure the scenario of event pausing, read the serial port and put data in queue.
	19 wait_for_decode	Start a timer to wait for decoding of data received to complete with timeout given
	17 start_receive_thread	start receive data thread, if any data receive, it will update the data
	18 toggle_data_receive	check the flag of pause event. If yes resume, if not pause.
	19 pause & resume_receiving_process	set the flag of pause receiving
	20 stop_receive_thread	stop receive data thread

	21 update_received_data	print the response at frontend & backend monitor
	22 clear_log_frontend	clear the log at front end data monitors
	23 open_port	read settings.ini config file and open port
	24 Wait_for_button press	Prompt a window to force and wait user to press reset button on board
	25 Force_close_button_press_window	Prompt mode select window when user close window while selecting config file, or close the prompt window while not detecting button press
	26 Close_button_press_window	Close the button press window when detect button being pressed
	27 closed_port	stop receiving thread & close port
	28 open_backend_window	get the password entry, and compare password. added all widget
	29 update_ports	update ports
	30 clear_log_backend	clear the log at back end data monitors
	31 send_data	send data manually
	32 process_incoming_data	send 1B and prompt listening mode
	33 open_configuration_window	open config window
	34 close_prompt_window	enabled back interaction with main window when closed

	35 close_find_config_file_window	Close the window that select config file, and return focus to main window
	36 force_close_config_file_window	Close the window that select config file, and prompt user back to mode select window
	37 close_backend_window	close backend window
	38 close_configuration_window	close configuration window
	39 close_command_panel	close command panel
	40 close_port_connection_window	close port connection window
	41 load_serial_settings	read serial settings from config file, if not create one
	42 create_serial_settings	create serial config settings and config file
	43 confirm_exit	add confirmation when close the main window

4
Front End - GUI

Features Introduction



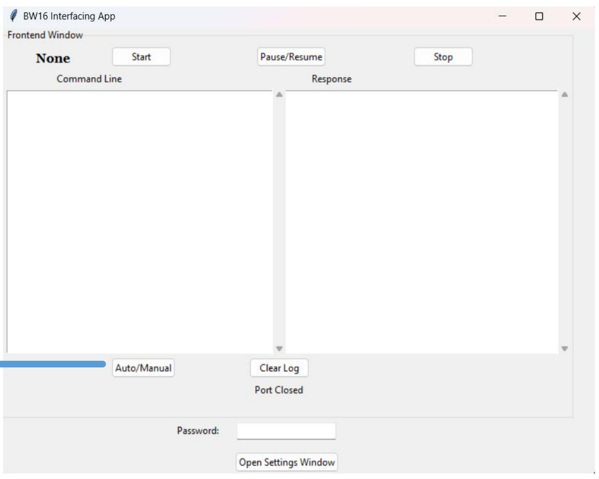
Id	Features	Function
1	Operating mode	Indicate user of current operating mode
2	Start	Auto: Automate the task based on config file loaded. Manual: Open port
3	Pause/Resume	Pause the automation and
4	Stop	Terminate the process and close the port
5	Auto/Manual	Enable user to select auto/manual operating mode
6	Clear log	Clear the log in both command line and response monitor
7	Open Settings Window	Prompt settings(backend) window if user input correct password.

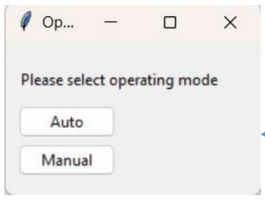
Monitor	Function
Command Line	Display the command sent to the serial port
Response	Display the feedback response from BW16

4
Front End - GUI

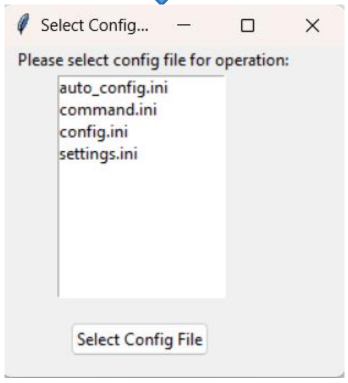
<u>Application Process</u>

Auto Mode





Please select operating mode

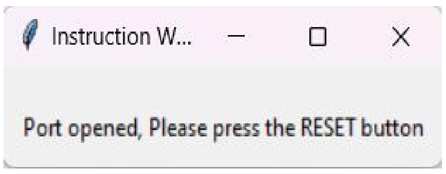


Please select config file for operation:

auto_config.ini
command.ini
config.ini
settings.ini

1

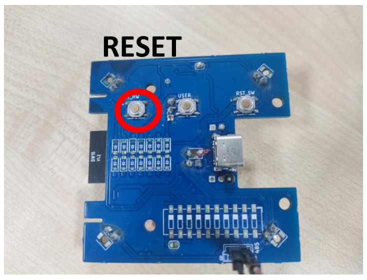
The list shows the config file available in the same directory with the application file.

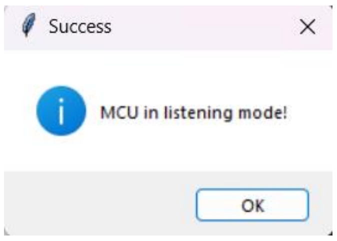


Port opened, Please press the RESET button

2

Selected config file will be loaded into GUI. The serial port will open and prompt the instruction window asking user to press "RESET" button on IR blaster after select config file





MCU in listening mode!

3

After pressing "RESET" button, an indication window will indicates user that the MCU is in listening mode.

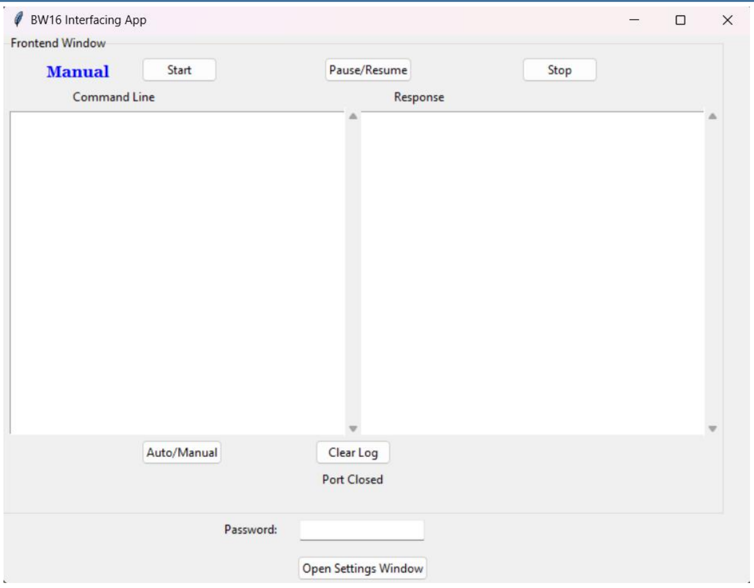
User can automate the command by "start" button, or manually input any command from backend window and get the response from MCU.

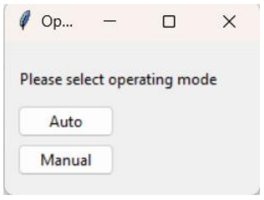
4
Front End - GUI

Application Process

Manual Mode

Manual

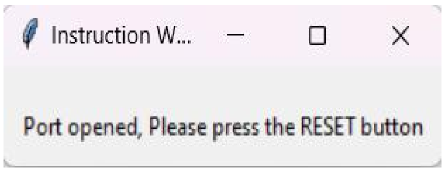




Please select operating mode

1

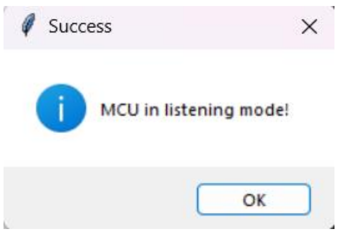
Pressing Manual button will make application enter manual mode. (It will clear the config file loaded if previously the GUI in auto mode and have any config file loaded)



Port opened, Please press the RESET button

2

Pressing "Start" button on front end window will prompt instruction window, asking user to press "RESET" button.



MCU in listening mode!

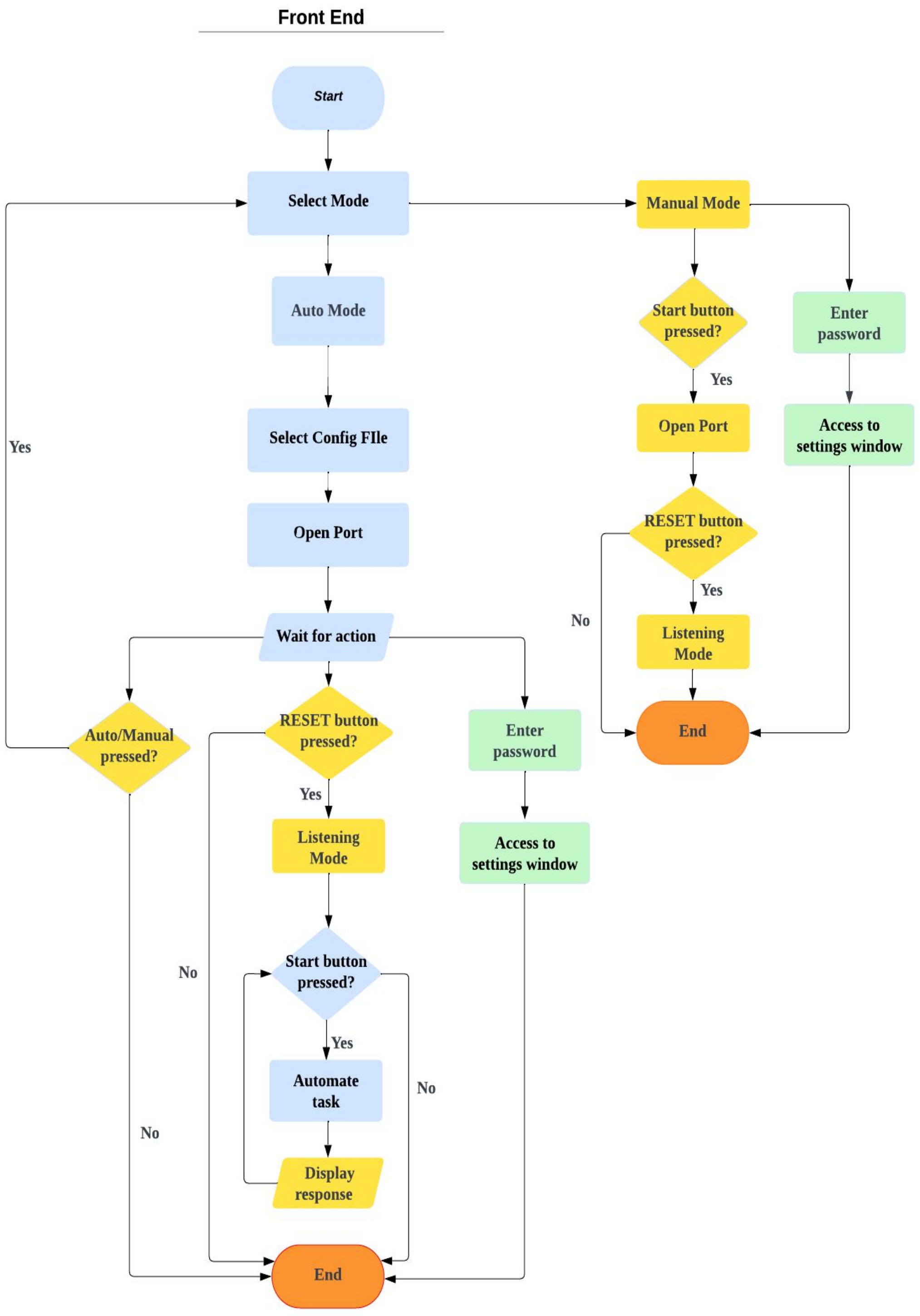
3

After pressing "RESET" button, an indication window will indicates user that the MCU is in listening mode.

4
Front End - GUI

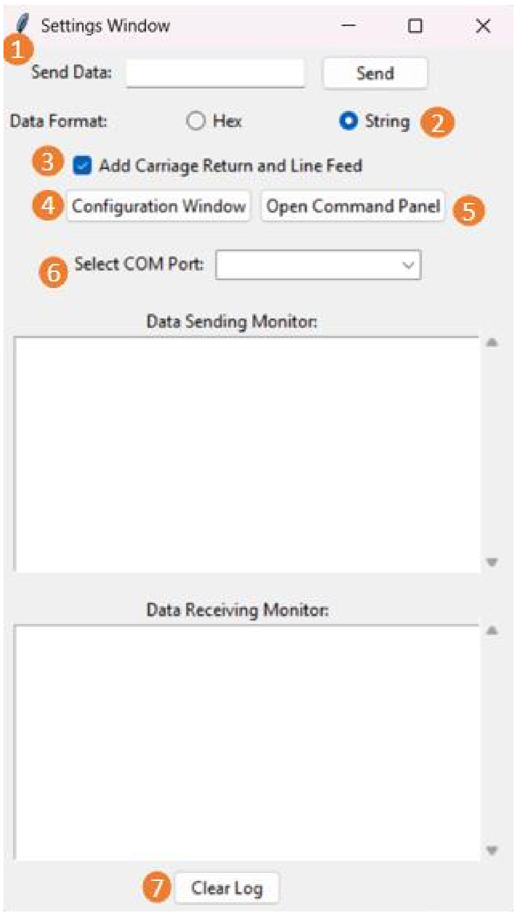
Flowchart

The flowchart demonstrates the overview of how the workflow of the application in the front end panel should operates. It consists of two operating mode which have different behaviors.



5
Back End - GUI

Features Introduction



Id	Features	Function
1	Send Data	Send the data manually
2	Data Format	Enable user to select data format to be sent
3	Add Carriage Return and Line Feed	Append carriage return and line feed to the data sent
4	Configuration Window	Open configuration settings window to change the config settings
5	Command Panel	Open command panel which consists of all commands
6	Com Port	Display the list of COM Port available which connected to serial device
7	Clear Log	Clear log on both monitor

5
Back End - GUI

Features Introduction

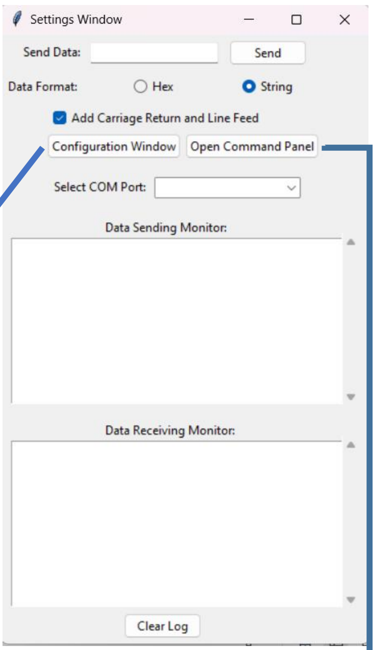
Configuration Window

Select the desired baud rate, data bits, stop bits, parity.

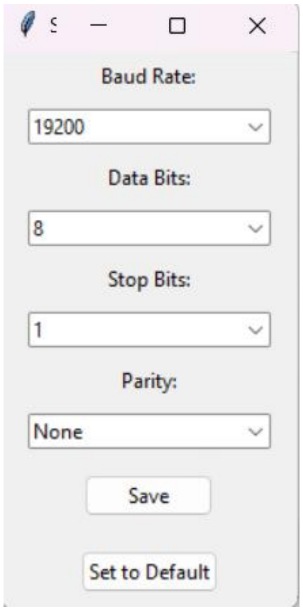
Save - save the configuration settings and apply it.

Set to default - Set the configuration settings to predefined default settings.

Settings Window



Data Format: String



Baud Rate: 19200

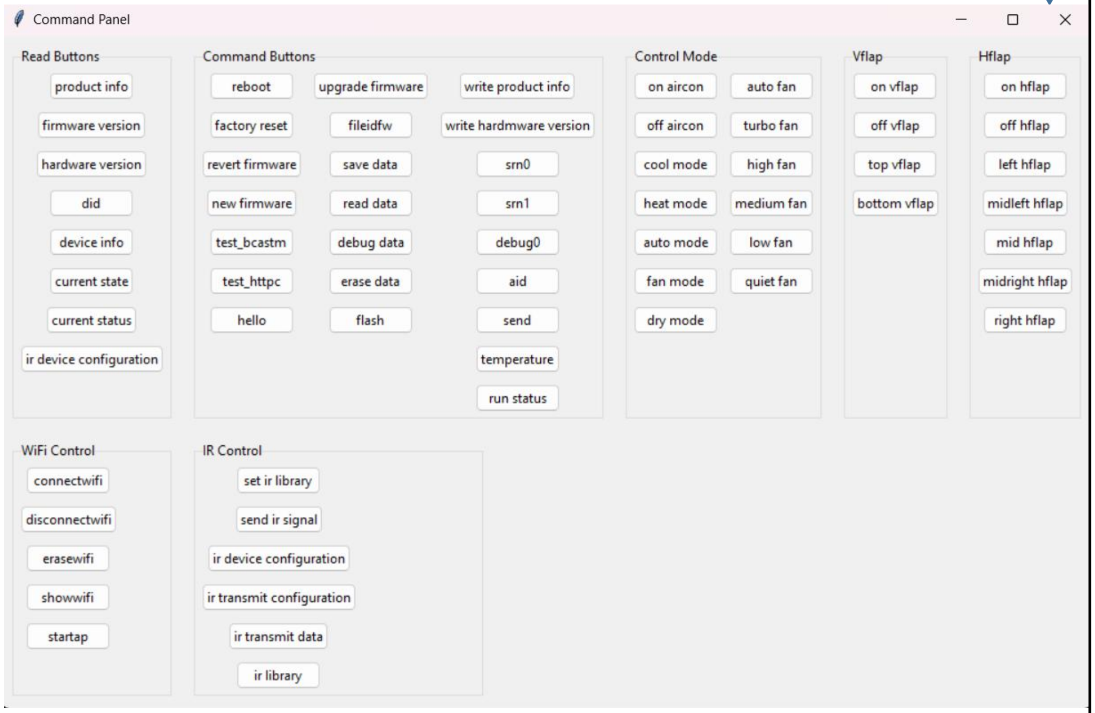
Data Bits: 8

Stop Bits: 1

Parity: None

Command Panel

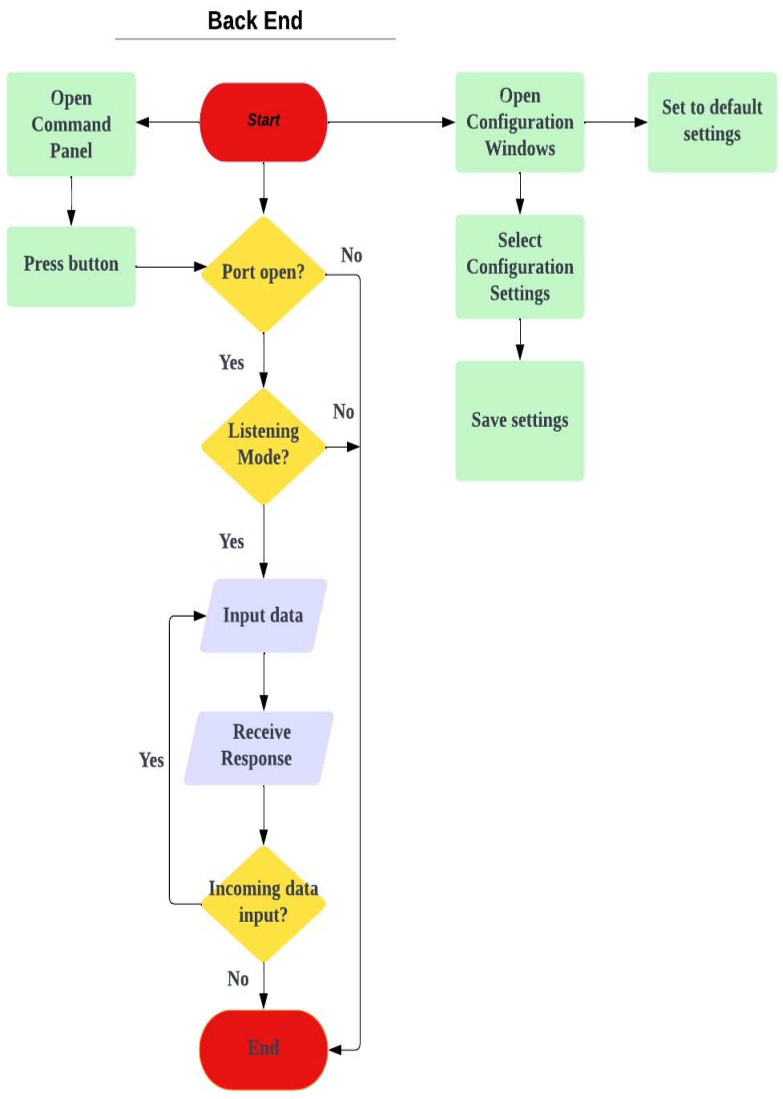
Command Panel consists of all categorized command used to interface with IR blaster.



5
Back End - GUI

Flowchart

This flowchart introduces all the features and behavior of back end window.



6
Config file list

<u>List of config file used</u>

1. Auto_config

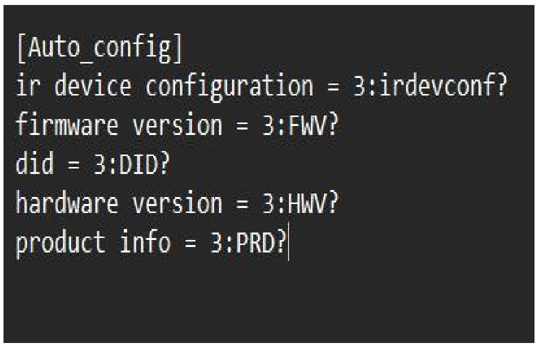
2. Command (auto create)

3. Settings (auto create)

4. Config (auto create)

Auto_config

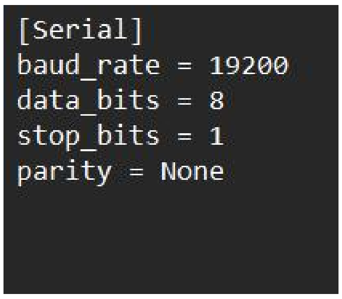
Config file for automate the task



```
[Auto_config]
ir device configuration = 3:irdevconf?
firmware version = 3:FWV?
did = 3:DID?
hardware version = 3:HWV?
product info = 3:PRD?
```

Settings

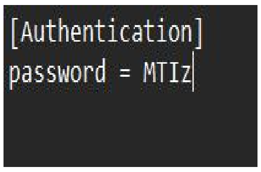
Store the configuration settings.



```
[Serial]
baud_rate = 19200
data_bits = 8
stop_bits = 1
parity = None
```

Config

the entry password undergoes base64 password encoding and will saved as config file located same directory as executable file.



```
[Authentication]
password = MTIz
```

Command

Consists of all command name and its corresponding data value to be send.

7
Executable Files Compile and Creation

<u>Pyinstaller</u>

Pyinstaller bundles a Python application and all its dependencies into a single package. User can run the packaged app without installing a Python interpreter or any modules.

When creating executable files, pyinstaller will find all import statements in script and create a list of all modules and libraries used. It will creates standalone executable file that includes script, dependencies, and python interpreter.

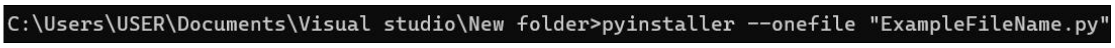
1. Create executable files with terminal

- Locate to where the location is of the script file
- Type the key command to create executable files.

<u>Key Command</u>

pyinstaller --onefile “ExampleFileName.py”

Example:



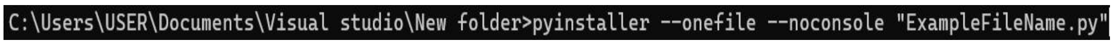
```
C:\Users\USER\Documents\Visual studio\New folder>pyinstaller --onefile "ExampleFileName.py"
```

2. Create normal executable files (without terminal)

- Locate to where the location is of the script file
- Type the key command to create executable files.

<u>Key Command</u>

Pyinstaller --noconsole --onefile “ExampleFileName.py”



```
C:\Users\USER\Documents\Visual studio\New folder>pyinstaller --onefile --noconsole "ExampleFileName.py"
```

8
Future development

1. Provide Software Update for GUI

2. <u>GUI features update plan</u>

- Additional button for downloading automated test script
- Provide test criteria for every command control
- Handle the scenario where multiple devices connected

3. Bugfixes & Failsafe mechanism

Handle the situation where any unexpected error occurs.

Consider all potential scenarios that might happen to crash the program and provide ways to overcome the scenario.

4. Provide instruction sheet as a guide for user to use the program.

9
Progress Update

Outline	Features / Functionality
Completed	● Automated data sending ● Different operating modes serve for different purposes and behavior. ● Serial Data Communication with IR Blaster ● Password encoding ● Indication window for different situation ● Compile and categorize all commands into a single panel
Work in Progress	● Bugfixes ● Enhance the pause/resume behavior ● Apply data streaming for IR Blaster automation ● Improve the UI design and indication ● Handle the behavior when multiple devices connected.
Planning stage	● Provide software update for GUI program ● Provide a feature for user to download automated test script ● Provide test criteria for every command ● Add instruction sheet to guide user way to use the program