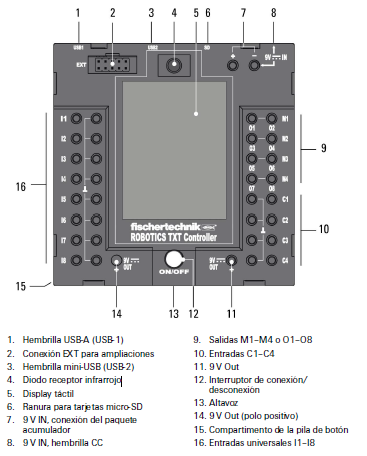
|  |  |
| --- | --- |
|  | AMBIENTE INTEGRADO DE APRENDIZAJE  DEPARTAMENTO DE INGENIERÍA INDUSTRIAL |

**MANUAL DE CONEXIÓN TXT CONTROLLER**

**Caracterización del Controlador**

A continuación se muestra una imagen del controlador donde se explican sus partes así como entradas y salidas.



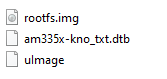
La salida que se usará principalmente será la número 2, la cual permite la ampliación para el uso del protocolo I2C para los motores. Adicionalmente, se pueden usar las salidas O1-O8 para LEDs, sensores, etc. Las salidas y entradas de voltaje serán a 9V.

La distribución del software de este controlador está construida sobre Builtroot, que es una distribución embebida de Linux usando compilación cruzada. Este puede soportar algunas librerías de C y por medio de elementos como el Community Firmware se puede correr aplicaciones de Python.

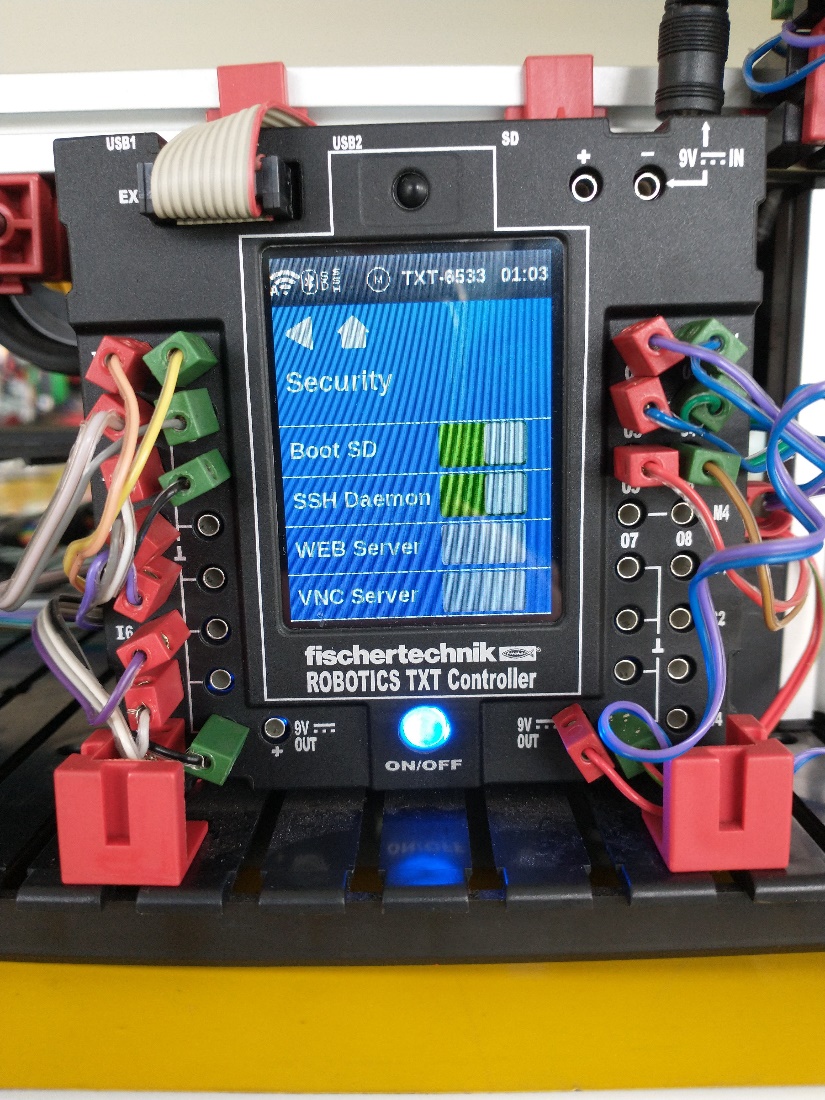
**Community Firmware**

A continuación se explica el uso del TXT controller por medio de Python, utilizando el *Community Firmware (*[*https://cfw.ftcommunity.de/ftcommunity-TXT/en/*](https://cfw.ftcommunity.de/ftcommunity-TXT/en/)*).* Para el uso de este, la versión del TXT Controller tendrá que ser superior a 4.2.4. Adicionalmente, se necesita una tarjeta de memoria micro SD formateada completamente (formatear usando la aplicación *SD Card Formatter*). Los pasos para la instalación del firmware son:

1. Descargar el Community Firmware (CF) de <https://github.com/ftCommunity/ftcommunity-TXT/releases/tag/v0.9.5>
2. Insertar en la micro SD únicamente los archivos de la carpeta comprimida descargada

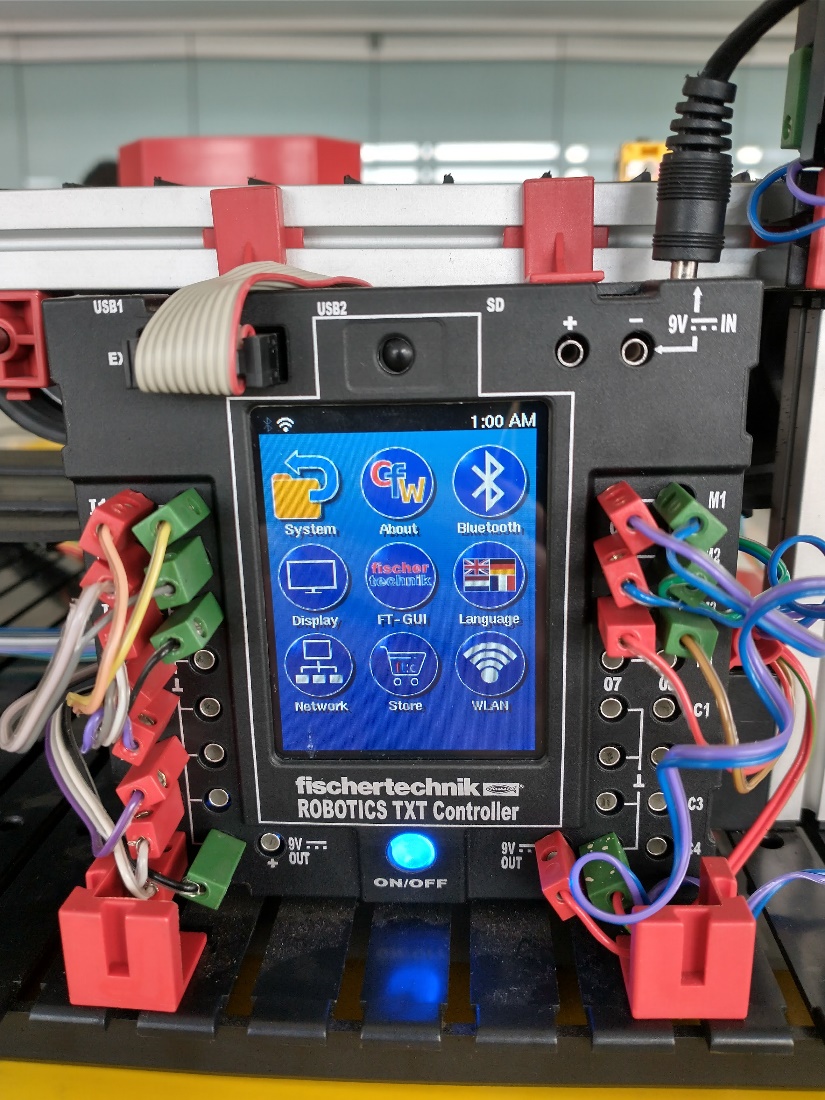


1. Habilitar “Boot SD” en el TXT Controller siguiendo la ruta *Settings > Security > Boot SD*



1. Apagar el TXT Controller
2. Insertar la memoria SD en el TXT Controller
3. Encender el TXT Controller

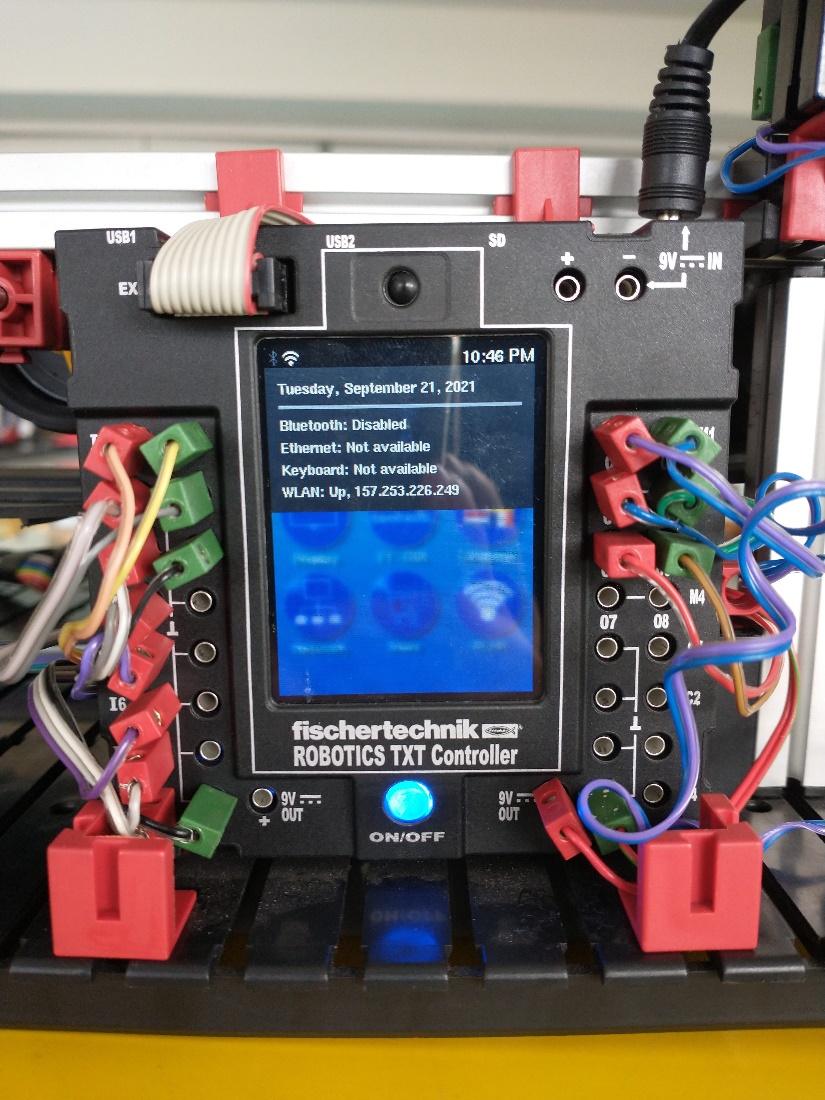
El TXT Controller deberá iniciar con el CF. Este tiene una apariencia diferente y más amigable con el usuario.



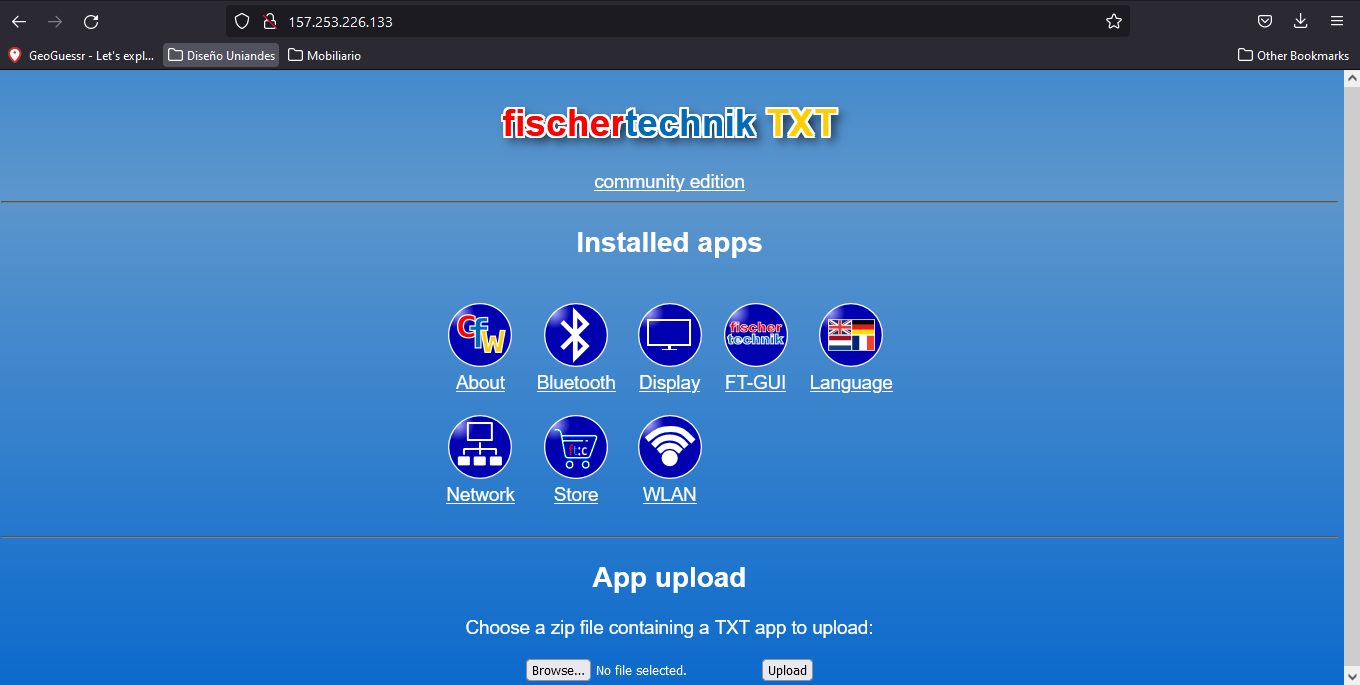
Una vez instalado el firmware, se procede a conectar el controlador a la Red de WiFi local (SSID: *AIALAB, Pass: LabAIA2021*).



Luego de conectado, presionando la parte negra superior se debe poder visualizar la dirección IP del controlador.



Ingresando a esta dirección desde un computador conectado al mismo *Access Point,* se tendrá acceso a la interfaz gráfica del CF. Desde esta interfaz se podrá cargar programas al TXT Controller.

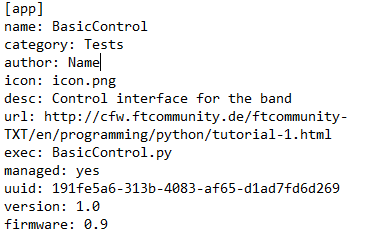


**Diseño de aplicaciones**

Para el diseño de aplicaciones, se parte de la guía de aplicaciones de CF (<https://cfw.ftcommunity.de/ftcommunity-TXT/en/programming/python/tutorial-1.html>). Esta guía indica que para cualquier aplicación es necesario generar 3 archivos:

1. Manifest: Este archivo debe contener la descripción de la aplicación a usar, integrando información como nombre, archivo a ejecutar, ícono, etc. (ver ejemplo)

.



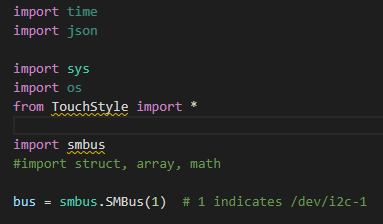
Se puede generar el UUID desde este vínculo (<https://www.famkruithof.net/uuid/uuidgen>). Es muy importante que sea diferente para cada aplicación.

1. Icon: El ícono debe adjuntarse en formato JPG o PNG y tendrá un tamaño de 64x64 pixeles.
2. Aplication: La aplicación deberá estar construida sobre Python y debe tener una interfaz gráfica básica para poder ejecutarla.

A continuación se detalla la forma de implementar las aplicaciones

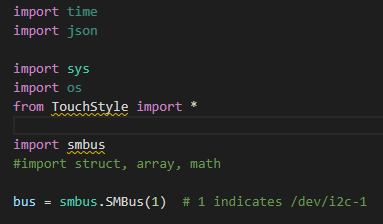
1. **Librerías**

Es importante integrar las librerías necesarias para operar las bandas desde el controlador. Las mas importantes serán ***Touchstyle***y***os***para el control gráfico, ***Ftrobopy*** para el control general y ***smbus***para el control de los motores (I2C). No es necesario tener estas librerías en el computador, ya que se ejecutan directamente desde el TXT Controller.

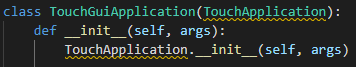


1. **Definiciones**

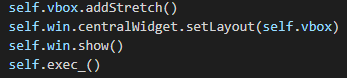
Es importante definir ***bus***como se muestra a continuación, ya que sobre este se escribe la salida del I2C para el control de los motores.



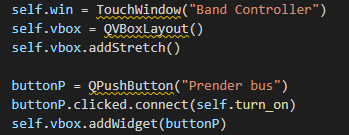
También se debe definir la interfaz del programa. Para esto se define la clase ***TouchGuiApplication*** que se muestra a continuación. Dentro de esta clase se definen los elementos de la interfaz y las funciones que se usarán en el programa.



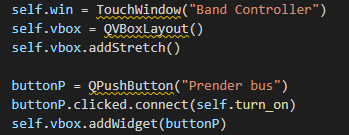
Dentro de la función ***init*** se debe incluir la siguiente sentencia, que permite inicializar la ventana y mostrarla en el TXT Controller.



La ventana se define con la función ***TouchWindow*** que recibe como parámetro el nombre de la aplicación. Luego se configura el atributo ***vbox*** como se muestra en la imagen a continuación



Finalmente, se pueden agregar objetos en la interfaz como se muestra en la imagen inferior. Es importante definir la función del botón que se agrega y que esta función esté definida dentro de la clase ***TouchGuiApplication.***



1. **Control**

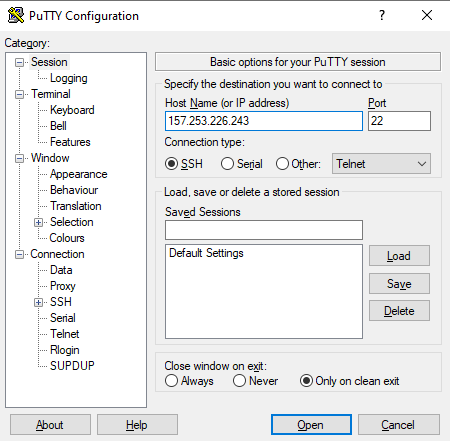
Para controlar las bandas se usa la función ***write\_byte*** de la librería ***bus***. Esta función recibe sus parámetros como valores **hexadecimales** y se le debe indicar el puerto de salida y el valor del mismo (velocidad). Los valores se definen para que coincidan con la configuración del Arduino usado para el I2C, por lo que el puerto de salida siempre será el puerto 8 y los valores para la velocidad estarán entre 0 y 10.



1. **SSH**

La ejecución del código se puede realizar por medio de SSH, el cual se puede conectar por medio de una aplicación como **PuTTy**. Es importante destacar que esta sección es completamente opcional y no es necesaria; sin embargo, si es necesario en caso tal que se desee realizar un *debug* sobre la aplicación creada o revisar algún *output* desde el computador que controla la banda. A continuación se muestra el procedimiento para la conexión.

1. Ingresar la IP del controlador en PuTTY con el puerto por defecto y especificando SSH en el tipo de la conexión. Luego presionar *Open.*



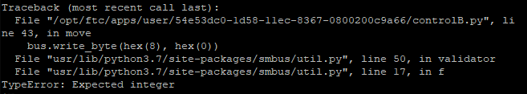
1. Una vez se abra la ventana correspondiente de manera automática, se debe ingresar el usuario para login. El usuario de las bandas es ***ftc*** y no tiene contraseña.



1. Se debe ingresar el comando *export PYTHONPATH=/opt/ftc* y luego se debe indicar la ruta del archivo a ejecutar, la cual estará dentro de la carpeta *apps* con el UUID que se asignó previamente y además el archivo de Python a ejecutar.



En caso de encontrar algún error, la consola mostrará en dónde se encuentra y porqué se dio:



1. Finalmente, se puede utilizar el comando *nano* para editar el archivo de Python desde la consola, guardándose automáticamente en el TXT Controller sin necesidad de volverlo a cargar por la aplicación.
2. **Paramiko**

Una alternativa para usar SSH desde Python es usar la librería Paramiko. El funcionamiento de esta librería es muy sencillo y se puede usar de la siguiente forma:

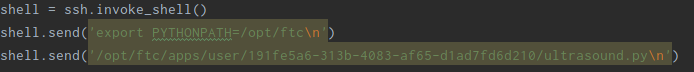
* + - 1. Se crea un cliente SSH y se le asigna una autoverificación de *lost keys*, para indicar que se confía en la conexión.



* + - 1. Crear la conexión al TXT con la IP, el usuario *ftc* y la contraseña vacía



* + - 1. Crear una conexión al *shell* y enviar los comandos del SSH por medio de esta. Los comandos a enviar son los mismos que se envían por PuTTy



1. **Documentación varia**

* Ejemplo de código de uso de las bandas: *BasicControl.py*
* Community Firmware: <https://cfw.ftcommunity.de/ftcommunity-TXT/en/>
* Documentación smbus/ smbus2: <https://buildmedia.readthedocs.org/media/pdf/smbus2/latest/smbus2.pdf>
* Buildroot: <https://buildroot.org/>
* DatasheetTXT Controller: <https://www.google.com/url?sa=i&url=https%3A%2F%2Fcontent.ugfischer.com%2Fcbfiles%2Ffischer%2FZulassungen%2Fft%2FTXT-Controller_es.pdf&psig=AOvVaw1KMn89D92WAE4pn-Ab5jjT&ust=1634164086411000&source=images&cd=vfe&ved=2ahUKEwiR5KLg9cXzAhVMZ98KHRv2CrEQr4kDegQIARA3>