Migrating MakeCode Target from micro:bit to i.MX8ULP

This document describes the full process of creating and configuring a custom MakeCode target for the NXP i.MX8ULP board, based on the micro:bit target.
It includes all file changes, build steps, and debugging insights.

1. Prerequisites

- Before starting please refer https://hub.akhilalabs.com:85/akhilalabs/Nocode/src/branch/master/Documentation/NO_CODE%20Documentation.odt and make sure you have installed pxt-microbit according to it.
- Copy pxt-microbit and name it as pxt-imx.

2. Modifications for i.MX8ULP Target

- Update pxtarget.json
  Edited target.json to reflect the new board.

  -Refer the pxtarget.json file.
  - Removed micro:bit-specific options such as yottaTarget, codalTarget, and deployDrives.
    Added dockermake build engine.
  - Add docker-build.sh
    Created/modified the docker-build.sh script:
  - Adjusted appTheme

- Update package.json
  -Refer the package.json file.

3. Adding and Removing files

- Add files inside pxt-imx/
- docker-build.sh
- Dockerfile

4. Building the Target

- From inside pxt-imx:
  $ sudo rm -rf built/ sim/
  $ sudo pxt buildtarget
  $ sudo pxt serve
- Expected output:
  Using target imx8ulp with build engine dockermake
  target.json built.
  bundling editor...

Starting server in /home/user/makecode/pxt-imx
starting local ws server at 3233…

To launch the editor, open this URL:
http://localhost:3232/#local_token=...

Open the given URL in browser → GUI loads.

Adding a Custom Block to PXT Target (IMX8ULP)

This document explains how to add a custom block named **showAkhila** to your PXT target so that it appears in the MakeCode Blocks GUI.

1. Edit libs/core and Add Function

Refer the libs/core folder of your target and locate the file that contains the namespace basic(basic.ts).

Explanation:

- namespace basic — This ensures the block appears in the **Basic** category.
- //% blockId=basic_show_akhila — Registers a unique block ID so Blockly can render it.
- block="show Akhila" — Text that will appear on the block in the editor.
- weight=95 — Determines order (higher weight = higher position in toolbox).
- console.log — Will print to browser console. Replace with showString() if your board supports display output.

2. Edit pxt.json inside libs/core

- Refer pxt-imx/libs/core/pxt.json

3. Rebuild Target

After editing the file, rebuild your PXT target:

```
$ sudo pxt clean
$ sudo pxt buildtarget
```

This step regenerates target.json, target-strings.json, and updates cached API info.

4. Reload the Editor

Run the local server if it's not already running:

```
$ pxt serve
```

Open the editor URL (e.g., http://localhost:3232) in your browser. Create a new project and check the **Basic** category. You should now see a new block named **show Akhila**.