PXT MakeCode for i.MX: Using Custom Docker for TypeScript to Binary Conversion

1. Folder Structure Example

auto_docker/
├── auto_build.js
├── basic.cpp
├── con.js
└── docker.sh
├── Makefile

---

2. Flow

auto_build.js

STEP 1: Clone or pull repo
STEP 2: Check if .ts file exists
STEP 3: Copy .ts to transpiler working dir
STEP 4: Run transpiler (con.js)
STEP 5: Move generated main.cpp to repo dir
STEP 6: Execute docker.sh which creates final_bin using Makefile and copies that into imx terminal using scp

con.js

1. Read TypeScript input
2. Parse TypeScript into AST
3. Shim map (TS → C++)
4. Emit C++

3. Steps

- After creation of project upload it to github
- After the upload is successful run this command -

  $ node auto_build.js

This is show this message
Pulling latest changes...

From https://github.com/()

 * branch   master   -> FETCH_HEAD

Already up to date.

Copied main.ts to transpiler directory.

Running transpiler...

Generated main.cpp copied to repo folder.

Automation complete.

Cleaning build artifacts...

aarch64-linux-gnu-g++ -Wall -c main.cpp -o build/main.o

aarch64-linux-gnu-g++ build/main.o -o binary/final_bin

final_bin
100%   21KB   1.6MB/s   00:00

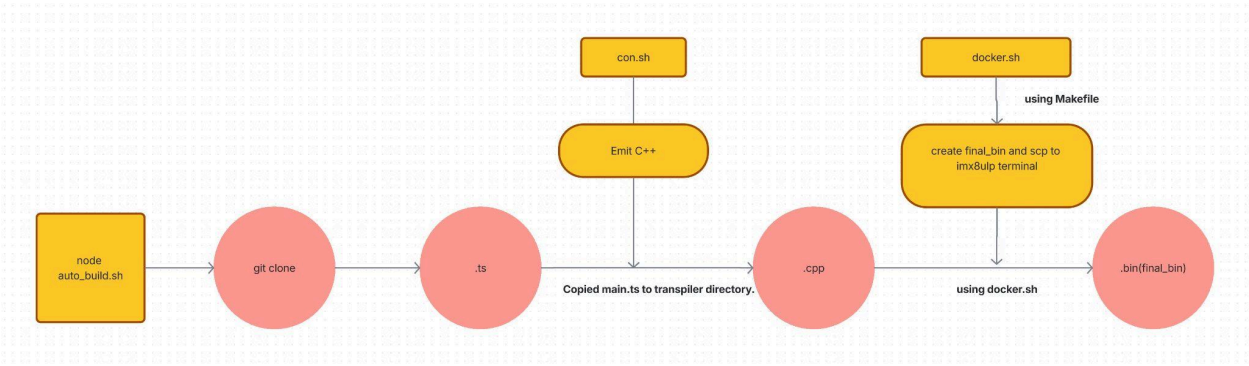
Shell script executed successfully!

You can see that main.cpp and final_bin file is created

- Inside imx8ulp terminal

  Look inside /nocode/final_bin

- Run it $./final_bin (observe the output)


con.sh
docker.sh
using Makefile
Emit C++
create final_bin and scp to imx8ulp terminal
node auto_build.sh
git clone
.ts
Copied main.ts to transpiler directory.
.cpp
using docker.sh
.bin(final_bin)