

Caravel Nucleo Hat

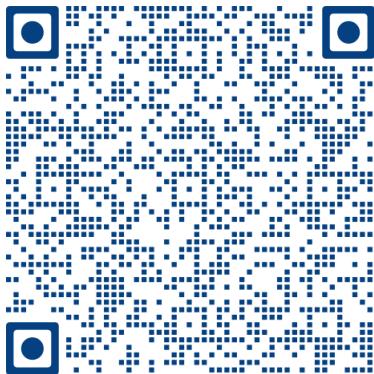
This directory provides a diagnostic software for characterizing timing failure patterns between GPIO pads on Caravel for the MPW-2, MPW-3 and related shuttles.

The diagnostic runs on a STM Nucleo development board in combination with a Caravel Hat board that hosts the Caravel part under test.

The current version of this document can be found at

https://github.com/efabless/caravel_board/blob/main/firmware_vex/nucleo/README.md

or scan the QR code...



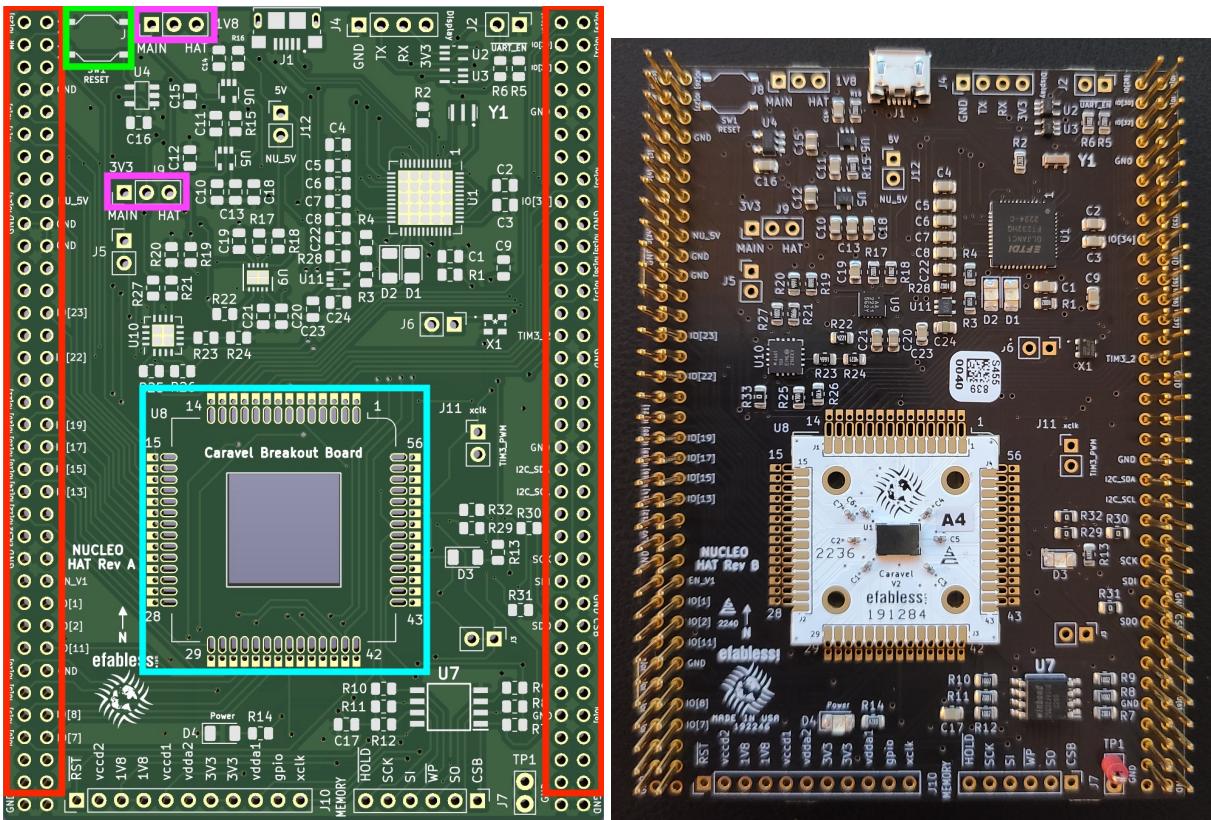
Setup

COMPONENTS

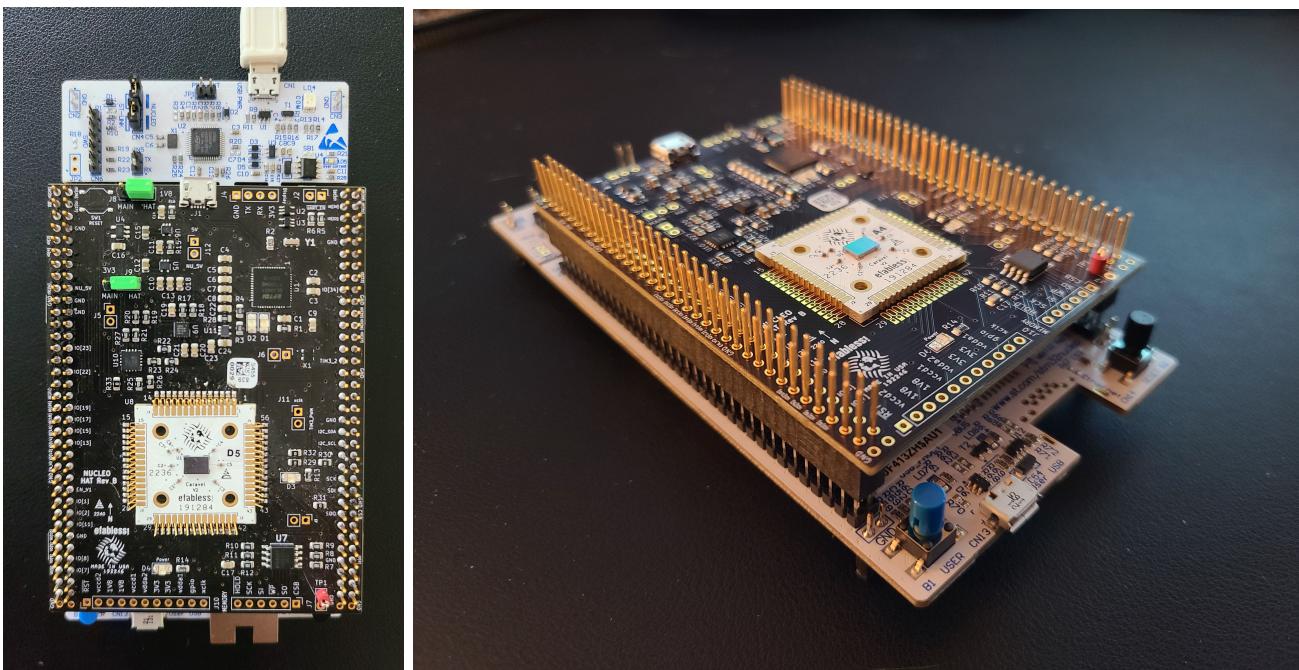
- NUCLEO-F746ZG or NUCLEO-F413ZH
- Caravel Nucleo Hat
- One or more Caravel breakout boards with a Caravel part installed
- Two jumpers for J8 & J9
- USB micro-B to USB-A cable

CONFIGURATION

- Install the jumpers on J8 and J9 in the 'HAT' position to enable the board to be powered by the Nucleo.



- Plug the Caravel Nucleo Hat in Nucleo board pins
 - the USB on the hat should face the ST-LINK breakoff board on Nucleo and away from the push buttons on Nucleo



- Install a Caravel Breakout board into the socket on the Caravel Hat board
 - the Efabless logo should face the USB connector on the Hat

- Connect the USB to a workstation / laptop
- Clone the github repo https://github.com/efabless/caravel_board.git
- Change to the firmware_vex/nucleo directory
- Run `pip install mpremote`

To run the diagnostic

```
git clone https://github.com/efabless/caravel_board.git

cd caravel_board/firmware_vex/nucleo

pip3 install mpremote

cd firmware_vex/nucleo

make run
```

The test will begin with the green LED on the Nucleo flashing 5 times.

When the test concludes, the green and red leds will be as follows:

GREEN	RED	STATUS
2 short + 4 long	off	Full Success - BOTH IO chains configured successfully
2 short	2 long	Partial Success - LOW IO chains configured successfully
2 short	4 long	Partial Success - HIGH IO chains configured successfully

If the test completed for the part, run the following to retrieve the configuration file. The file will indicate the IO that were successfully configured. Successfully configured IO can be used for this part for firmware routines.

```
make get_config
```

Using the Configuration File

... coming soon