

# GSoC 2022 Final Report: CAN Framework and Peripheral Support

## CAN Framework and DCAN peripheral support

This report summarises the project that I have worked on RTEMS CAN Framework and DCAN peripheral support throughout the Google Summer of Code 2022.

## Final commits

All the feature-specific commits are squashed into two commits (CAN framework and DCAN support).

- [CAN Framework support](#)
- [DCAN peripheral support](#)

## Project Overview

The objective of this project is to add CAN Framework and DCAN peripheral support for BeagleBone Black System on Chip in RTEMS. CAN protocol is a robust, reliable and multi-master serial communication protocol used to achieve real-time message transfer between devices within the CAN network. RTEMS being a real-time operating system, CAN peripheral support would increase the potential to meet real-time demands.

## Project Objectives

- Add CAN Framework.
- Add DCAN peripheral support.

## Mentors

- Christian Mauderer
- Pavel Pisa
- Gedare Bloom

## Summary

In this project, we have implemented CAN Framework with Tx, minimal Rx support and DCAN peripheral support for BeagleBone Black in RTEMS. I have summarized the work that I have done in each phase.

Project Proposal: [Google Doc](#)

Project Blogs: [Blogs](#)

Gitlab Repo: [Gitlab Repo](#)

- [DCAN Peripheral Patch](#)
- [CAN Framework Patch](#)

## Phase 1

In the first phase,

- Design CAN Framework.
- CAN Hardware test setup ([Communication between two BeagleBone Black through CAN bus](#)).
- Adding DCAN driver to RTEMS in loopback mode.

## Phase 2

In the second phase,

- Implementing CAN framework.
- Adding test applications.
- Adding Tx and Rx FIFO in CAN framework.
- Adding Concurrency handling mechanisms.
- Adding DCAN interrupt support.

## Phase 3

In the third phase,

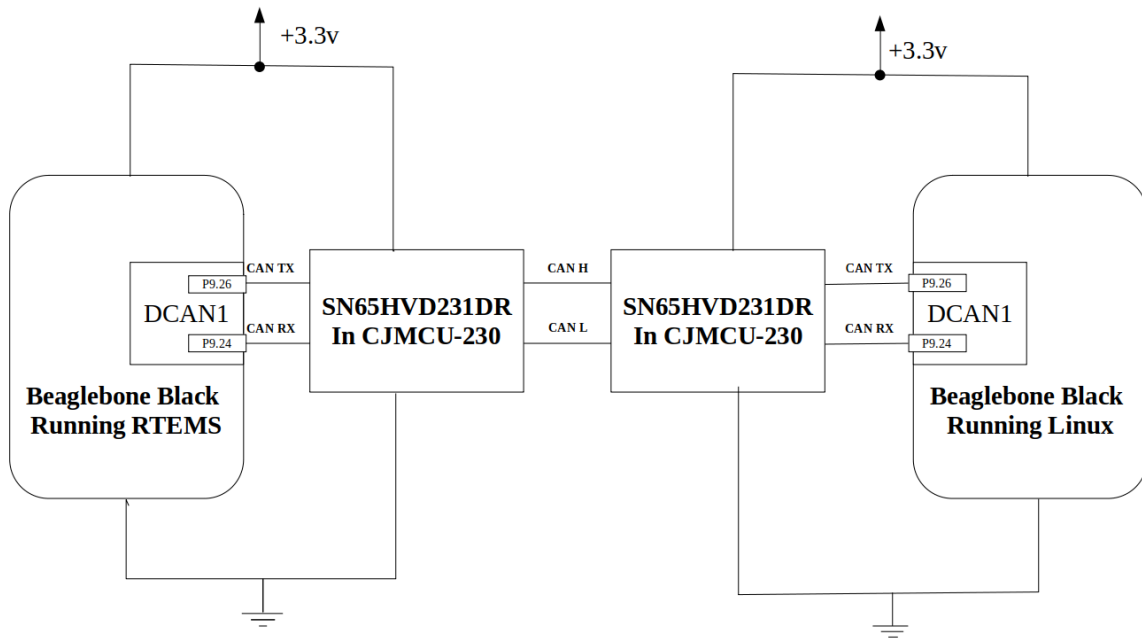
- Integrating DCAN and CAN framework.
- Adding CAN loopback driver.
- Testing CAN framework and DCAN driver.
- Adding documentation.

## Output

### CAN Loopback Test Application

[CAN-Test-application](#) is the test application which creates a CAN loopback interface and does Tx and Rx on the file.

### Output on Beaglebone Black



This figure shows the hardware setup used for testing CAN between two BeagleBone Blacks.

```

root@beaglebone:~# sudo config-pin p9.24 can
.26 can

sudo ifconfig can1 down
sudo ip link set can1 up type can bitrate 1000000
sudo ifconfig can1 up

Current mode for P9_24 is:    can

root@beaglebone:~# sudo config-pin p9.26 can

Current mode for P9_26 is:    can

root@beaglebone:~#
root@beaglebone:~# sudo ifconfig can1 down
root@beaglebone:~# sudo ip link set can1 up type can bitrate 1000000
root@beaglebone:~# sudo ifconfig can1 up
root@beaglebone:~#
root@beaglebone:~# q
-bash: q: command not found
root@beaglebone:~# candump can1
   can1  00000001  [6] 30 00 00 00 31 00

```

This image shows a CAN message sent from the BeagleBone Black application running on RTEMS with the CAN framework, which is received by another Beaglebone Black running Linux.

## Acknowledgement

I like to thank my mentors **Christian Mauderer**, **Gedare Bloom** and **Pavel Pisa** for their guidance and support throughout my project. I like to thank **Oliver Hartkopp** for the initial guidance in designing CAN Framework.

I also like to thank the **RTEMS community** (**Joel Sherrill**, **Karel Gardas** and many more) for the opportunity and support throughout the project.