GSoC Proposal for BeagleBoard.org

BeagleBoard.org
Jul 15, 2025

Table of contents

Chapter 1

Introduction

Introducing Zephyr RTOS support for the BeagleBone AI-64 platform, with the aim of upstreaming it to the Zephyr repository. Our focus will be on enhancing driver support for essential communication protocols on the TDA4VM SoC. Additionally, we plan to implement inter-process communication (IPC) mechanisms to facilitate message queuing between the A72 core and R5 cores, thereby enabling efficient communication and coordination within the system. This endeavor seeks to empower developers with a robust real-time operating system foundation and comprehensive hardware support for embedded applications on the BeagleBone AI-64 platform.

1.1 Summary links

- **Contributor:** Dhruv Menon
- **Mentors:** Dhruva Gole, Nishanth Menon, Andrew Davis
- **Code:** TBD
- **Documentation:** TBD
- **GSoC:** NA

1.2 Status

This project is currently just a proposal.

1.3 Proposal

- Created accounts across OpenBeagle, Discord and Beagle Forum.
- The PR Request for Cross Compilation: #184
- Created a project proposal using the proposed template.

1.4 About

- **Forum:** u/malto101 (Dhruv Menon)
- **OpenBeagle:** melta101 (Dhruv Menon)
- **Github:** malto101 (Dhruv Menon)
- **School:** Manipal Institute of Technology

- **Country:** India
- **Primary language:** Englsih
- **Typical work hours:** 8AM-5PM Indian Standard Time
- **Previous GSoC participation:** N/A

Chapter 2

Project

Project name: Upstream Zephyr Support on BeagleBone AI-64 R5

2.1 Description

The Beaglebone AI 64 platform is a formidable tool for developing embedded systems, Zephyr OS support will allow it to reach even greater potential. Zephyr is a lightweight real-time operating system that improves embedded applications' scalability, flexibility, and dependability.

Part of the Beaglebone AI 64 platform, the System-on-Chip (SoC) has an intricate architecture that includes two Arm Cortex-A72 cores and four Arm Cortex-R5F cores. An addditional two Arm Cortex-R5F cores are located on a different MCU island. Such configurations epitomize asymmetric multiprocessing (AMP), facilitating the concurrent operation of a full-fledged operating system (OS) alongside a real-time operating system (RTOS).

The suggested approach uses *remoteproc* from the Linux-powered A72 core to load the Zephyr RTOS onto the BeagleBone AI 64. Upstreaming Zephyr support for the BeagleBone AI 64 platform is the main goal of this project, which builds on the foundation set by the previous Google Summer of Code (GSoC) contributor which already includes crucial support for features like the VIM interrupt controller, DM timer for systick, and Davinci GPIO controller on the J721E SoC. Defining Kconfig based existing board support and using the current Hardware Model v2(HWMv2) instead of previous OOT(Out of the Tree) model while Adding major drivers to allow seemless interaction such as I2C, SPI, mailbox. In addition, the project attempts to show seamless communication across cores by introducing example instances of Inter-Process Communication (IPC) for example, we could use message queues to synchronize events.

Defination

- **Vectored Interrupt Manager (VIM)**: interrupts to Cortex R5F are managed through VIM in TDA4VM.
- **DM Timer**: The DM timer module generates the system tick. The interupt is caused by overflowing, compare or capture.
- **Davinci GPIO controller**: The controller provides functionality for managing digital input and output signals, allowing the SoC to interact with external devices and peripherals.

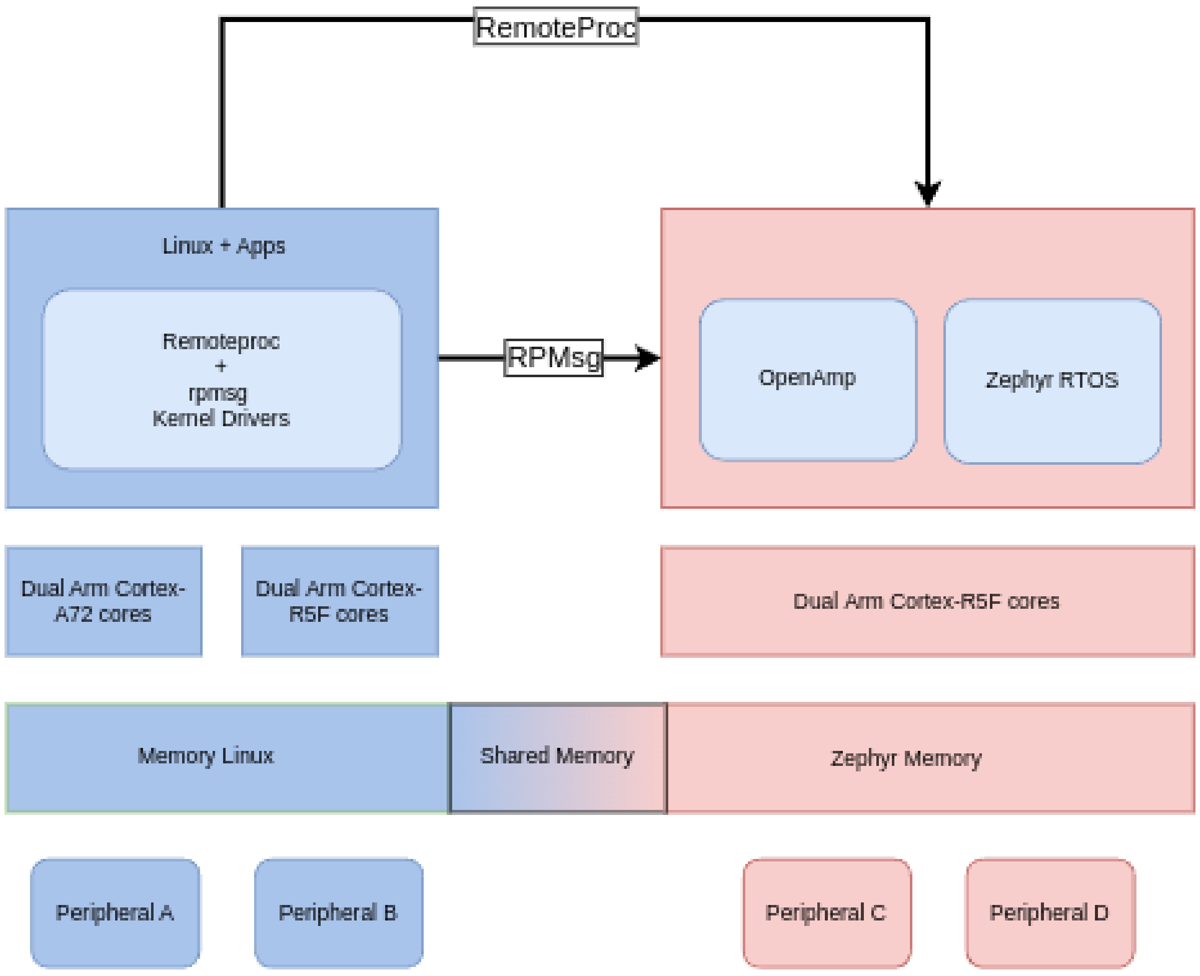
2.2 Motivation

1. Extend Platform capabilities by offloading some services(for example DSP) on the MCU.
2. Manage diversity of services on a single link.
3. Introduces fault isolation by segragating critical tasks onto dedicated cores.

2.3 But why Zephyr?

Zephyr RTOS (Real time Operating System) is a robust and flexible option for developing embedded systems, including real-time functionality, scalability, portability, security, flexibility, and a vibrant community. Zephyr offers the essential capabilities and tools to develop embedded applications effectively and dependably, regardless of the complexity of the device—from basic sensor nodes to intricate IoT devices. The Cortex R5 processor core present on the TDS4VM are built to provide deeply embedded **real-time and safety-critical systems**. Thus, adding Zephyr RTOS support for R5 cores in TDA4VM will be very helpful for the product developers.

Important: Zephyr recently began supporting AMP System-on-Chip (SoC) architectures, exemplified by the introduction of the Hardware Model v2 (HWMv2). This is significant as AMP architectures, like TDA4VM, feature heterogeneous cores with varying processing capabilities, allowing for efficient distribution of tasks between different cores based on their strengths.



[1]

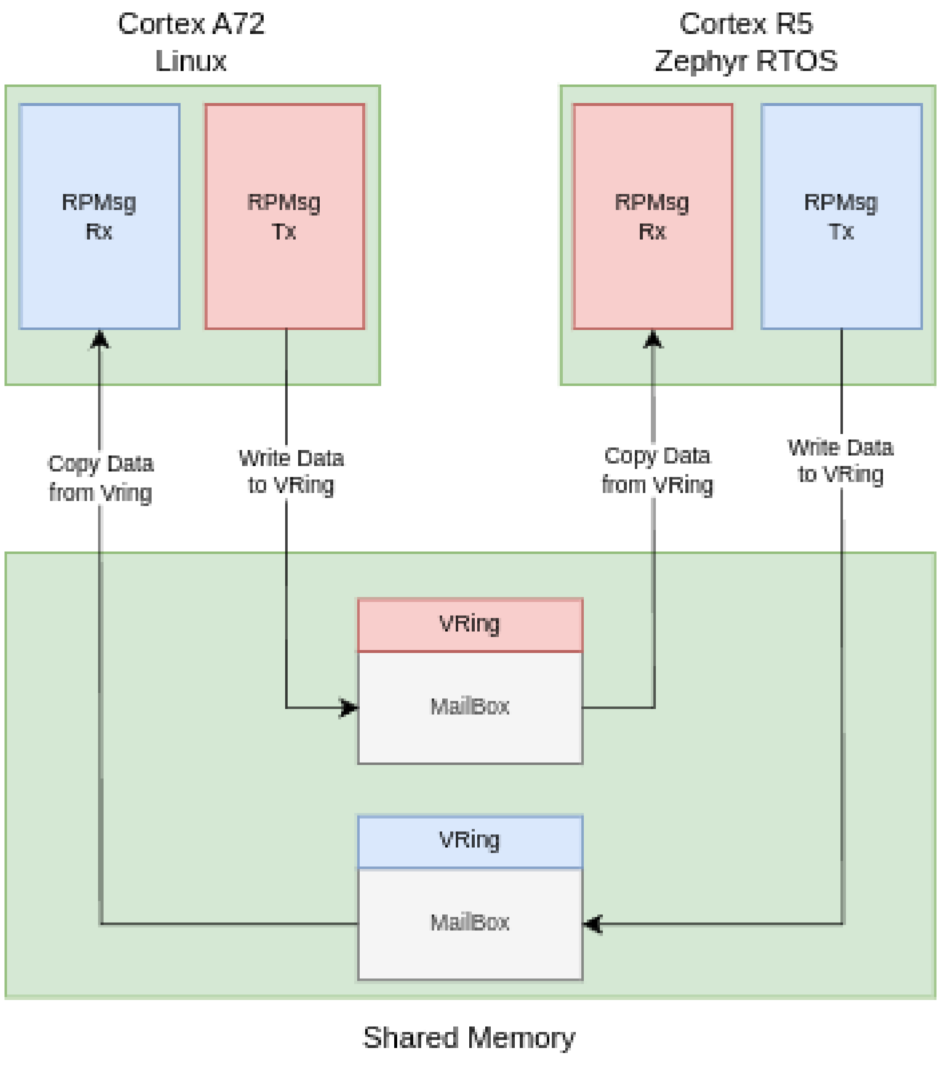
2.4 remoteproc

The remote processor (rproc) framework serves as a robust solution for managing remote processor devices within modern System-on-Chips (SoCs) and is particularly geared towards heterogeneous multicore processing (HMP) setups. This innovative framework provides the means to control various aspects of remote processors, such as **loading and executing firmware**, all while effectively abstracting the underlying hardware differences. Furthermore, it extends its utility by offering services for monitoring remote coprocessors, thereby ensuring management and operation.

The Linux running on the Cortex-A72 uses the remoteproc framework to manage the Cortex-R5F co-processor. Therefore, the binary to be copied to the SD card to allow the A53 cores to load it while booting using remoteproc.

2.5 Remote Processor Messaging (rpmsg) Framework

Typically AMP(Asymmetric Multiprocessing Processor) remote processors employ dedicated DSP codecs and multimedia hardware accelerators, and therefore are often used to offload CPU-intensive multimedia tasks from the main application processor. These remote processors could also be used to control latency-sensitive sensors, drive random hardware blocks, or just perform background tasks while the main CPU is idling. Rpmsg is a **virtio-based messaging bus** that allows kernel drivers to communicate with remote processors available on the system. In turn, drivers could then expose appropriate user space interfaces, if needed. is a message passing mechanism that requests resources through *remoteproc* and builds on top of the virtio framework. Shared buffers are requested through the resource_table and provided by the remoteproc module during Zephyr firmware loading



[2]

Defination

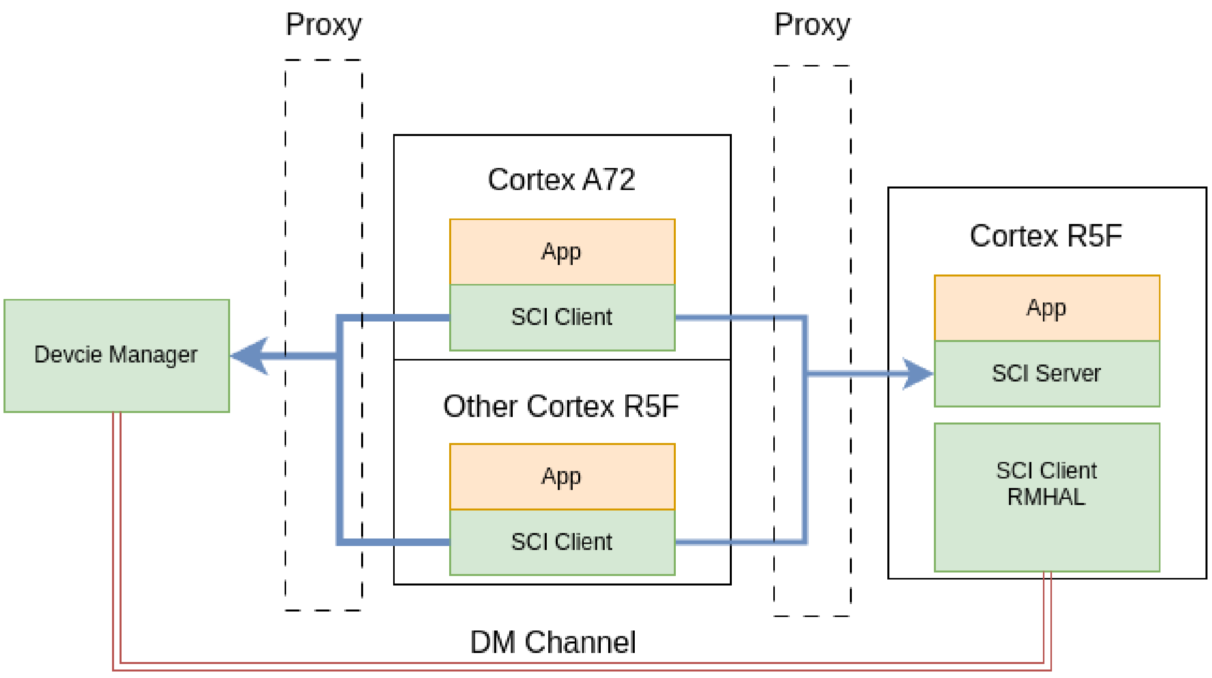
- **Vring**: SW queue that is based on shared memory and is used to keep messages while they are being exchanged between two CPUs.
- **HW Mailbox**: Hardware mechanism that notifies two CPUs of an interrupt

2.6 Open Asymmetric multiprocessing(OpenAMP) Framework

The OpenAMP framework provides software components that enable development of software applications for Asymmetric Multiprocessing (AMP) systems. The framework provides key capabilities such as **Life Cycle Management**, and **Inter Processor Communication** capabilities and Provides a stand alone library usable with zephyr environments Life Cycle Management (LCM) in OpenAMP is provided by the *remoteproc* component. Remoteproc APIs allow software applications running on the master processor to manage the life cycle of a remote processor and its software context while IPC(Inter Process Communication) a mechanism that allows processes to communicate with each other and synchronize their actions. The communication between these processes can be seen as a method of co-operation between them. Processes can communicate with each other through both: Shared Memory and Message passing

2.7 TI-SCI(Texas Instruments System Controller Interface)

SCI is one the primary communication protocol for a TDA4VM processor from which the host uses multiple threads to communicate with the systems firmware. This is a set of message formats and operation sequence required to talk to the system service in the SoC.



[3]

The Cortex-R5F core acts as the SCI server, which means it hosts the TISCI firmware responsible for managing system resources and peripherals while The Cortex-A72 core acts as the SCI client, interacting with the TISCI firmware running on the Cortex-R5F core.

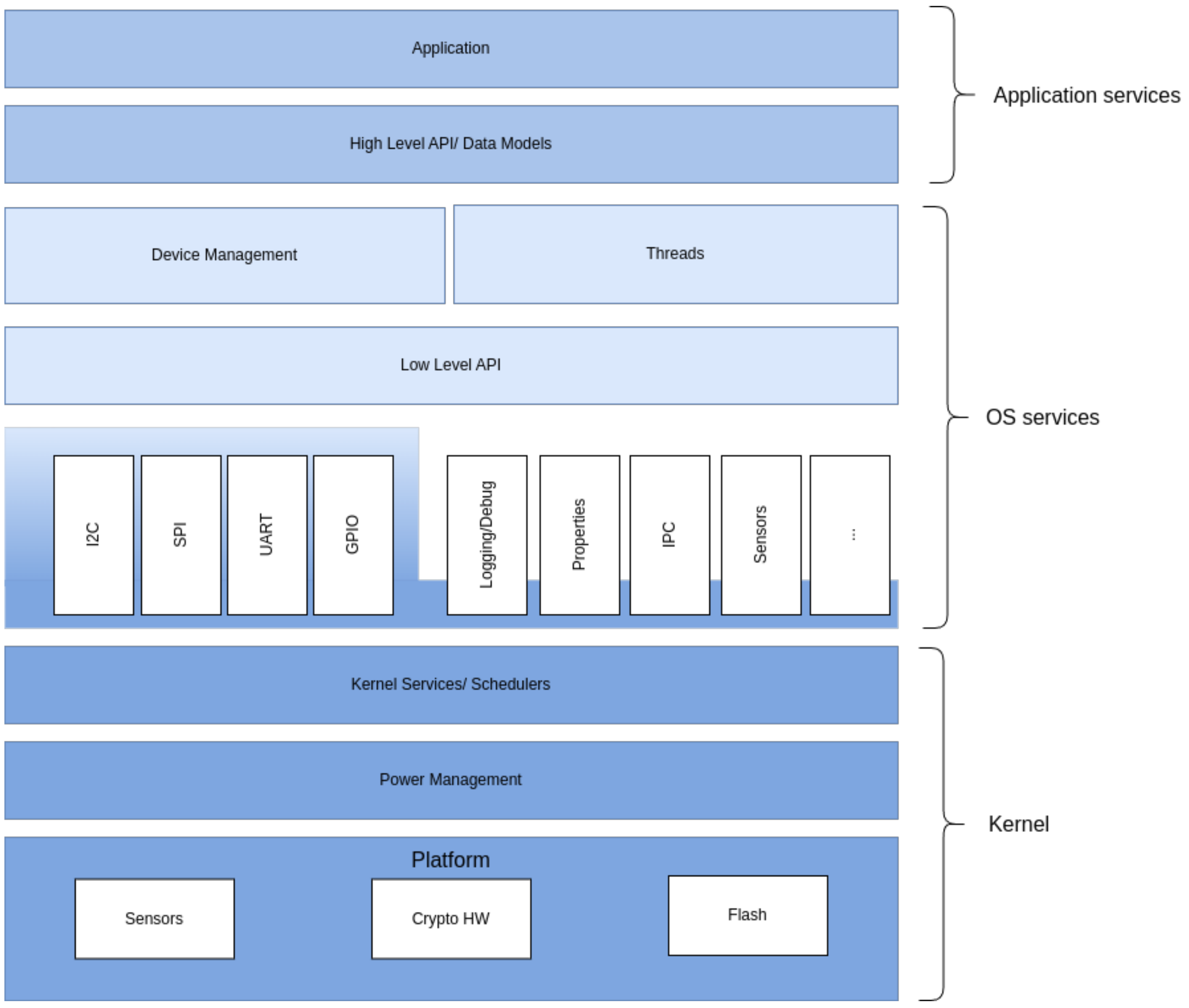
- Upon boot up, The SCI client (A72) will initialize the communication channels with the SCI server (R5F), ensuring bidirectional communication for sending commands and receiving responses.
- Client can request resource allocation, set clock frequencies, manage power domains, configure interrupt controllers, and control peripheral devices on the system.
- The Client can also query with the SCI server for status of the device.

Important: Upon implementation of SCI, we can add device management support to either control the state of a module or control the frequency of the clock to a module/(device reset control). In addition to the above messages, the TISCI also supports certain general messages.

The base scenario will be where MCU1_0 having the Zephyr RTOS will act as the SCI server and allow other client to communicate with it. TO begin, we will be providing the support for SCI client.

2.8 Expected System architecture

The scope of the project by the end is shown below:



2.9 Software

- C
- Zephyr RTOS

2.10 Hardware

- Beaglebone AI 64

Chapter 3

Timeline summary

Date	Activity
February 26	Connect with possible mentors and request review on first draft
March 4	Complete prerequisites, verify value to community and request review on second draft
March 11	Finalized timeline and request review on final draft
March 21	Submit application
May 1	Start bonding
May 27	Start coding and introductory video
June 3	*Milestone #1, Introductory YouTube video (June 3rd)*
June 10	*Milestone #2 (June 10th)*
June 17	*Milestone #3 (June 17th)*
June 24	*Milestone #4 (June 24th)*
July 1	*Milestone #5 (July 1st)*
July 8	*Submit midterm evaluations (July 8th)*
July 15	*Milestone #6 (July 15th)*
July 22	*Milestone #7 (July 22nd)*
July 29	*Milestone #8 (July 29th)*
August 5	*Milestone #9 (Aug 5th)*
August 12	*Milestone #10 (Aug 12th)*
August 19	*Final YouTube video (Aug 19th)*

Chapter 4

Timeline detailed

4.1 Community Bonding Period (May 1st - May 26th)

- Acquiring the Board and hardware ready.
- interaction with mentor for feedback.
- Mastering the hardware nuances and intricacies of the Zephyr codebase.
- look deeper into Prashanth.S commits

4.2 Coding begins (May 27th)

- Setting up Zephyr Environment.
- Flash the linux image on the A72 core.
- Boot the Cortex R5 firmware remotely from a Linux instance running on the A72 core using remoteproc.

4.3 Milestone #1, Introductory YouTube video (June 3rd)

- Include a introductory video.
- Write basic hello world on the board.
- Leverage resources from previous GSoC contributions to create the board directory . for the Beaglebone AI-64. This includes Kconfig configuration files and Device Tree (DT) definitions according to the HWMv2.
- Integrate west toolchain manager with the Zephyr build system for streamlined package management and compilation.
- Prepare Documentation.
- Add DM Timer support based out of Prashanth DM Timer.

4.4 Milestone #2 (June 10th)

- UART controller support in Zephyr for the SoC and bring up the shell through that. Configure the DT to enable the chosen UART port and integrate the driver with the board code. This will enable a serial console for debugging and interaction.
- Test using the UART echo sample.

- Add Support to GPIO controller based out of Prashanth Davinci GPIO controller.

4.5 Milestone #3 (June 17th)

- Add additional perpheral support for I2C, SPI.
- Test I2C, SPI using sensor examples from zephyr (For example: MPU6050 or use another MCU to interface with it)

4.6 Milestone #4 (June 24th)

- Develop message passing protocols, shared memory regions, and synchronization primitives.This framework should include functional device drivers, appropriate SoC configurations, and tested communication mechanisms ready for further development and refinement in subsequent milestones.

4.7 Milestone #5 (July 1st)

- Provide basic support for IPC between RTOS and the Linux subsystem by introducing the drivers and SoC config for IPC. Either using Message queues or the shared memory approach.

4.8 Submit midterm evaluations (July 8th)

Important: **July 12 - 18:00 UTC:** Midterm evaluation deadline (standard coding period)

- Clean up the documentation.

4.9 Milestone #6 (July 15th)

- Provide Mailbox(Secure Proxy) support for BeagleBone AI 64.

Defination

- **Mailbox**: A mailbox is a kernel object that provides enhanced message queue capabilities that go beyond the capabilities of a message queue object. A mailbox allows threads to send and receive messages of any size synchronously or asynchronously.
- **Secure Proxy Mailbox**: Texas Instruments own mailbox controller

4.10 Milestone #7 (July 22nd)

- Develop a Zephyr driver for the TI-SCI (System Control Interface) according to Zephyr's Hardware Abstraction Layer (HAL) standard with refernce to TI PSDK-J721E. This enables a more portable and reusable driver implementation.

4.11 Milestone #8 (July 29th)

- Provide Support for TI-SCI(System Control Interface)

4.12 Milestone #9 (Aug 5th)

- While adding SCI support, Include a interrupt routing support which can let Zephyr effectively handle hardware interrupts.

4.13 Milestone #10 (Aug 12th)

- Include support for the Arm Cortex-R5F to control clocks via the device manager using Ti-SCI.

4.14 Final YouTube video (Aug 19th)

- Submit final project video, submit final work to GSoC site and complete final mentor evaluation
- Provide few examples to help kickstart other member of the community.

4.15 Final Submission (Aug 24nd)

Important: **August 19 - 26 - 18:00 UTC:** Final week: GSoC contributors submit their final work product and their final mentor evaluation (standard coding period)

August 26 - September 2 - 18:00 UTC: Mentors submit final GSoC contributor evaluations (standard coding period)

4.16 Initial results (September 3)

Important: **September 3 - November 4:** GSoC contributors with extended timelines continue coding

November 4 - 18:00 UTC: Final date for all GSoC contributors to submit their final work product and final evaluation

November 11 - 18:00 UTC: Final date for mentors to submit evaluations for GSoC contributor projects with extended deadline

---

Chapter 5

Experience and approach

- Was successful in utilizing remoteproc to load firmware into the BeagleBone AI-64's R5 core.
- Part of the Deveopment for a custom AD9364 Transciever device driver in a Zynq MPSoC Evaluation kit.
- Successfully done board bring-up for mainly 32-bit MCUs, integrating crucial functionalities such as USB, UART, I2C, and SPI protocols.
- Developed device (character and user-mode) drivers for prominent platforms including Beaglebone, Nvidia Jetson, and Raspberry Pi.
- Freelanced a HID over GATT (HoG) firmware capable of mouse control controled through USB connectivity using **Zephyr RTOS**.
- Engineered a system enabling seamless communication between a QT application and GNU Radio back-end, leveraging a HackRF device and STM32MP1. This system facilitates dynamic adjustment of modulation schemes.

5.1 Contingency

Upon encountering a roadblock in my project without access to my mentor, I'll leverage online resources, documentation, and peer forums from both BeagleBoard and Zephyr community to troubleshoot and find a solution independently. I have also been in contact with people who have themselves upstreamed zephyr support for SBCs(Single Board Computer).

5.2 Benefit

- Zephyr being a popular RTOS option, can help users with BeagleBone AI-64 use Zephyr without touching the lower level code and concentrate on thier application
- Adding Zephyr support will also increase the interoperability with other devices and systems
- Operating RTOS and Linux side by side can lead to a wide range of application which are time constrained.

5.3 References

- [1] https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.openampproject.org%2Fdocs%2Fwhitepapers%2FIntroduction_to_OpenAMPlib_v1.1a.pdf&psig=AOvVaw2EJ_cAYMdy7vgQ-Zdrykff&ust=1711958394883000&source=images&cd=vfe&opi=89978449&ved=0CBUQ3YkBahcKEwiIq7ORhJ6FAxUAAAAAHQAAAAAQFQ
- [2] https://medium.com/@warrierabhinav/a-novel-communication-stack-for-a-heterogeneous-multicore-system-with-asymmet

- [3] https://software-dl.ti.com/tisci/esd/latest/1_intro/TISCI.html
- https://github.com/zephyrproject-rtos/zephyr/pull/59191
- https://github.com/zephyrproject-rtos/zephyr
- https://docs.zephyrproject.org/latest/kernel/services/data_passing/mailboxes.html
- https://software-dl.ti.com/jacinto7/esd/processor-sdk-rtos-jacinto7/07_02_00_06/exports/docs/pdk_jacinto_07_01_05_14/docs/userguide/jacinto/overview.html
- https://software-dl.ti.com/jacinto7/esd/processor-sdk-rtos-jacinto7/09_00_00_02/exports/docs/pdk_jacinto_09_00_00_45/docs/apiguide/j721e/html/index.html
- https://github.com/OpenAMP/open-amp/wiki