Librobotcontrol support for newer boards - Himanshu Kohale

BeagleBoard.org
Jun 09, 2024

Table of contents

Chapter 1

Introduction

Introducing librobotcontrol package support with newer boards.

1.1 Summary links

- **Contributor:** Himanshu Kohale
- **Mentors:** Jason Kridner, Deepak Khatri
- **Code:** TBD
- **Documentation:** TBD
- **GSoC:** NA

1.2 Status

This project is currently just a proposal.

1.3 Proposal

Completed all the requirements listed on the ideas page

- Created accounts on openbeagle , forum , discord.com/users/869908108565168198
- Source dive and get to know with packages and examples.
- The code for the cross-compilation task can be found, Submitted through the pull request :- pull request
- Proposal :- librobotcontrol support for newer boards

1.4 About

- **Forum:** u/himanshuk
- **OpenBeagle:** openbeagle.org/Himanshuk
- **Github:** github.com/Himanshukohale22
- **School:** Veermata Jijabai Technological Institute
- **Country:** India

- **Primary language:** English, Hindi, Marathi
- **Typical work hours:** 8AM-5PM Indian standard timeline
- **Previous GSoC participation:** N/A

Chapter 2

Project

Project name: librobotcontrol support for newer boards .

2.1 Description

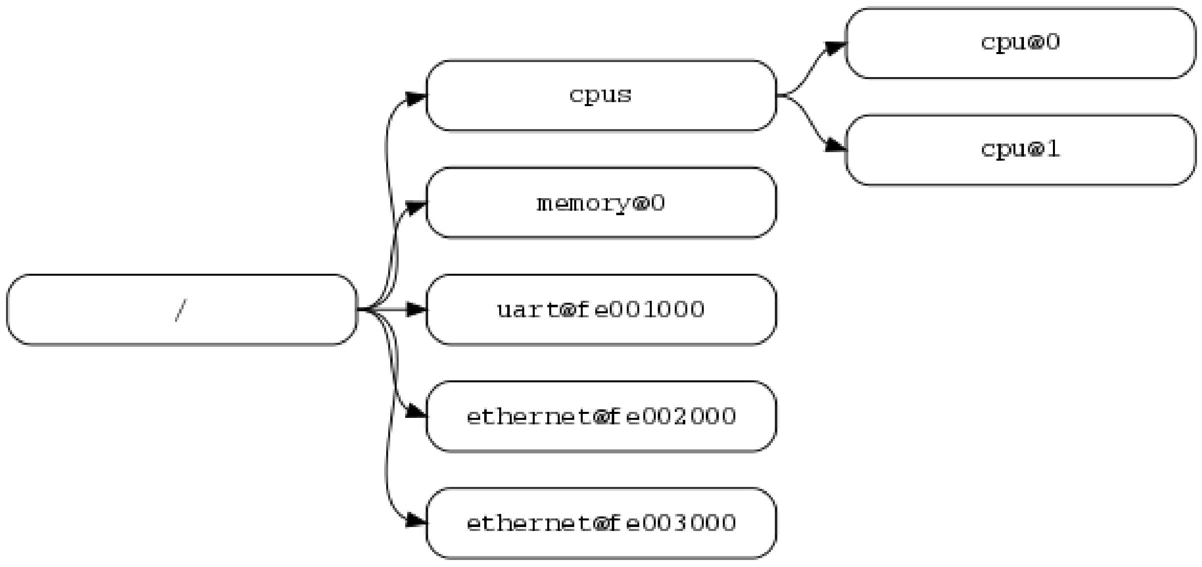
Overview

- Librobotcontrol is package of C library which contains examples and testing programms for Robotic control projects used by beaglebone capes like robotic-cape which is sold by beagleboard.org. BeaglBboneBlack(BBB)(am33xx) supports the librobotcontrol package thanks to Deepak khatri, Who Previouly worked upon cape compatibility layer on BBB.BeaglBboneBlack support robotic-cape cape with librobotcontrol package due to device tree overlays which identify the robotics-cape as a specific hardware which can be useful while accesing various pheripherals and devices with roboitic-cape.
- BeagleBone-AI support librobotcontrol package but its been draft and there is not a stable device tree overlays for robotic cape in AI image.So for using package, have to check the passed results for various drivers and use cape with AI and make changes accordingly. BeagleBone AI is based on the Texas Instruments AM5729 dual-core Cortex-A15 SoC with flexible BeagleBone Black header and mechanical compatibility. Which supports the AM572x device tree binary files. However, the current implementation has the following problems This proposal will address these issues.
- BeagleBone AI-64 uses TI J721E-family TDA4VM system-on-chip (SoC) which is part of the K3 Multicore SoC architecture. Each TI evm has an unique device tree binary file required by the kernel. As BeaglBboneBlack need (ti,am33xx) similarly beaglebone-AI64 support (ti,j721e) device tree binary (.dtb).BeagleBone AI-64 also support librobotcontrol packages but there are less tutorial and not refine code for support librobotcontrol package with new boards. Need to refine the device trees overlays to use this librobotcontrol package with AI-64.As in librobotcontrol there is no robotic-cape dtb support for beaglebone-AI64 we need to write the device tree overlays.
- BeagleV®-Fire is a revolutionary SBC powered by the Microchip's PolarFire® MPFS025T RISC-V System on Chip (SoC) with FPGA fabric. It has the same P8 & P9 cape header pins as BeagleBone Black allowing to stack BeagleBone cape on top to expand it's capability. Built around the powerful and energy-efficient RISC-V instruction set architecture (ISA) along with its versatile FPGA fabric. BeagleV-Fire also support the robotic cape with librobotcontrol package and cape gateware for robotic cape is pre-installed in V-fire image here . but as BBB support the librobotcontrol with more functionality and flexible application, beagleV-fire is not capable for librobotcontrol package due to less number of child node (PWM, GPIO) present in robotic-cape (.dts) file which is pre-installed in V-fire image. With help of customization for cape gateware in V-fire which Provide more flexibility to board with cape, and librobotcontrol package to support V-fire with robotic cape.
- Main goal of project is to update librobotcontrol package for beaglebone-AI(am5x), beaglebone-AI64(j721e) and beaeglV-fire(polarV-soc) boards. Primarily librobotcontrol support all the boards but not able to make roboitic-cape as flexible as BBB.

Implementation

- Device tree overlay are the data structure for Describing hardware. Rather than hard coding every details of a device into an operaing system, many aspect of hardware can be described in data structure that is passed to the OS at boot time.

Implementation of device tree :



- As above example, Root node is starting or begin the overall process of accesing hardware information after that there are CPU's, memory and various pheripherals which used as Nodes and sub-node are device-nodes where the information about specific hardware which will use that specific pheripherals is written.
- Similar to above example robotics capes need UART, I2C, SPI and GPIO's node. for librobotcontrol package.
- Have to write device tree overlays with corresponding pheripherals according to cape interface with beagleboards. Below is simple device tree example for accesing GPIO's for blink LED's with P8_7 and P8_8 headers pin which are inbuil-led's for roboitc-cape.

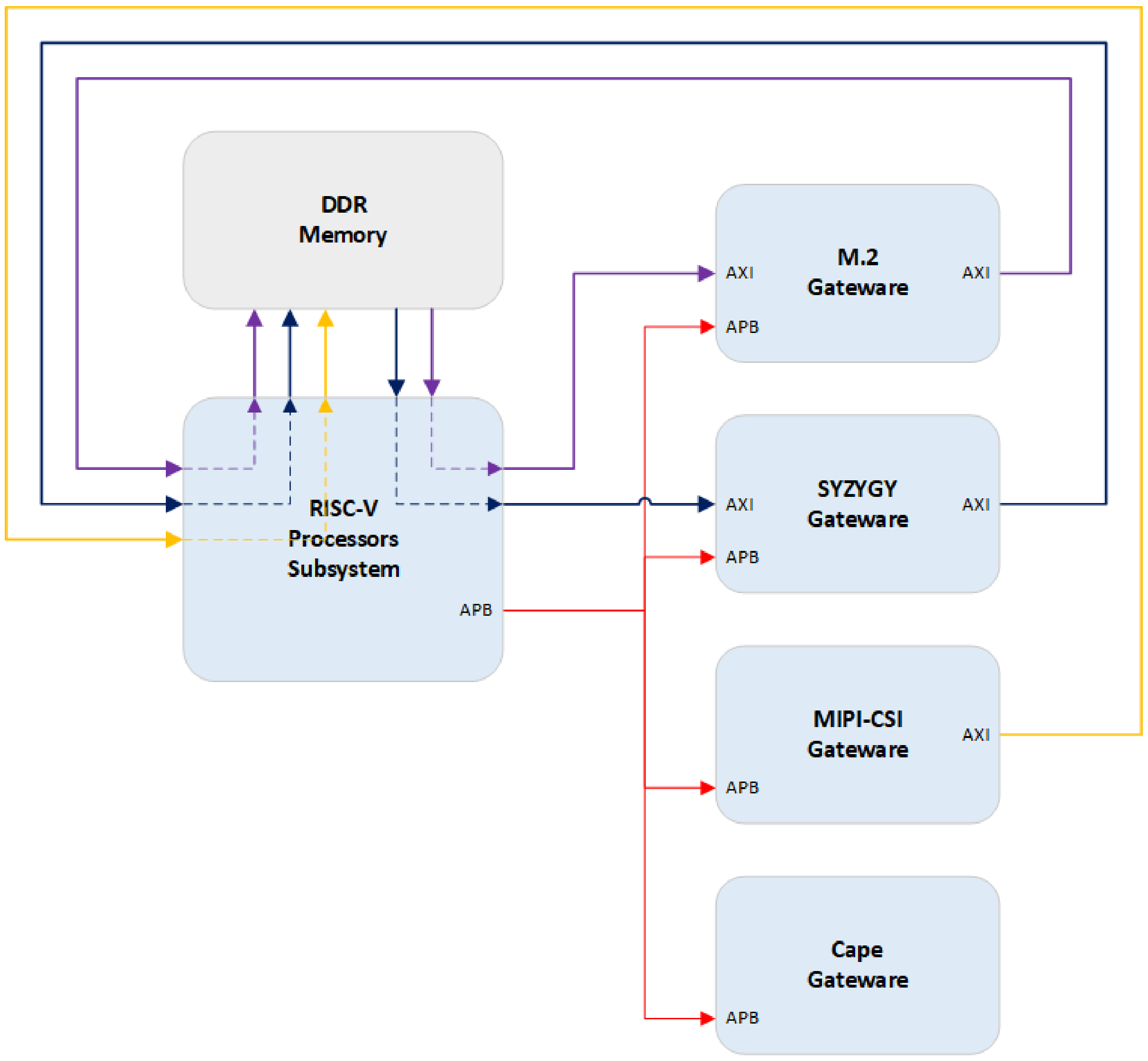
```
/{
    compatible = "ti,beaglebone-AI64";
    part-number = "LED_GPIO_TEST";
    version = "00A0";
    buildinLed@1{
        target = <ti,j721ex>;
        pinctrl-singl.pins = {
            0x090 0x0F      /*P8.7, MODE7*/
                        0x094 0x0F   /*P8.8*/
        }

    }

}
```

- After writing the source file for robotics-cape, This Robotic-cape source file for each board can be compiled using device tree compiler:

```
$ dtc -O .dtb -o robotic_cape.dtbo robotic-cape.dts
```

- In case of BeagleV-Fire boards, beaglV-fire support customization cape gateware which can be very useful to implement. Robotic_cape.dts source file is already present in V-fire image but with less specification and accesing PWM and GPIO's pheripherals which are bare minimum to use roboitc-cape with librobotcontrol. so for librobotcontrol use,have to update the device tree cape gateware.

V-fire Gateware architecture:



- Cape gateware in Gateware architecture of V-fire is responsible for handling the P8 and P9 connectors signals. The gateware is extended or customized by creating additional directories within the component directory of interest.

Previous work:-

Previously Deepak Khatri who worked upon the cape compatibility for beagleboards. use the robotic cape for various tasks. using pre-work upon robotic cape, i can take a deep dive to robotic cape compatibility with BeaglBboneBlack (BBB) and how its works. In previous gsoc-application 2022 participation kai yamada work upon same project which was about robotic-cape support with BeagleBone-AI (BB-AI). In both projects implementation was about the device tree overlayes for BBB and AI for specific pheripherals to enabling functionality of PWM, I2C and SPI and UART for robotic-cape.

2.2 Software

- Device tree's overlays for beagleboards will be used.The project requires the use of the device tree compiler (dtc) for compiling the device tree source (ex. *.dts, *.dtsi) files.
- Primarily VScode and gitlab with web-IDE is use in this project for deep dive into code and firmware of librobotcontrol and rc (robot control library) examples.
- C language.

2.3 Hardware

A list of hardware that you are going to use for this project.

- Beaglebone Black
- BeagleBone-AI
- Beaglebone AI 64
- BeagleV-fire
- **Beaglebone-capes**
  - Robotic cape
- **Additional hardware for project:-**
  - **Jumper cables :-**
    - 4-wire jst cables
    - 6-wire jst cables
  - DC motors
  - Servo motor
  - FTDI-TTL serial wire
  - SD-card
  - power supply 12v
- **Useful testing tools:-**
  - Oscilloscope
  - Multimeter
  - Soldering station
  - Mechanical toolbox

Chapter 3

Timeline

3.1 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	Release introductory video and complete milestone #1
June 10	Complete milestone #2
June 17	Complete milestone #3
June 24	Complete milestone #4
July 1	Complete milestone #5
July 8	Submit midterm evaluations
July 15	Complete milestone #6
July 22	Complete milestone #7
July 29	Complete milestone #8
August 5	Complete milestone #9
August 12	Complete milestone #10
August 19	Submit final project video, submit final work to GSoC site and complete final mentor evaluation

3.2 Timeline detailed

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

- Get to know with community, Read resources for librobotcontrol and beagleboards, get up to speed to begin working on the projects.
- At current period of time, all the required hardware will be available.
- Setup all the beagleboard hardware (Flashing OS and test hello world).
- Check all hardware with beagleboard like DC motors, Servo motors and available sensors.
- Use robotic-cape with beagleboard BeaglBboneBlack (BBB) and librobotcontrol.
- Use robotic-cape with BeagleBone-AI.
- Because my end-semester exams are starting, my availability for contributing will be limited.

3.2.2 Coding begins (May 27th)

- Given the onset of my end-semester exams, my ability to contribute will be reduced. Thank you for your understanding.
- Understand device tree overlays for BeaglBboneBlack (BBB) and AI written for robotic cape.

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

- Due to my End-semester Exam i won't able to contribute more, thank you for understanding.
- Include introductory video.

3.2.4 Milestone #2 (June 10th)

- End of exam and I can start to contribute more in project.
- Start to write Device tree for GPIO's and PWM support for AI-64.
- Test a device tree overlay to allow AI-64 to light the power LEDs with GPIO support.
- Test PWM Device tree overlay with robotics-cape with help of Hardware specification and check with oscilloscope.
- Get feeback from mentors.

3.2.5 Milestone #3 (June 17th)

- Write I2C node device tree for AI-64.
- Test I2C with IMU.
- Get feedback from mentor.

3.2.6 Milestone #4 (June 24th)

- Create merge request for I2C Device tree node.
- Write SPI device tree overlay for AI-64.
- Test with robotic-cape.
- Get feedback from mentor.

3.2.7 Milestone #5 (July 1st)

- Create RoboticsCape.dts file for robotic-cape which will support AI-64 using pre-work.
- Test .dts file with robotic cape with AI-64.
- Test example of librobotcontrol with AI-64.
- get feedback from mentor.
- Create merge request for RoboticsCape.dts.

3.2.8 Submit midterm evaluations (July 8th)

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

3.2.9 Milestone #6 (July 15th)

- Test RoboticsCape with cape gateware for beagleV-fire pre-installed in image.
- Understand the customization process for cape Gateware.

3.2.10 Milestone #7 (July 22nd)

- Customized LED example for robotic-cape gateware.
- Test GPIO’s, Robotic cape with beaglV-fire.
- Create merge request for LED blink with beaglV-fire.

3.2.11 Milestone #8 (July 29th)

- Examine SPI support for beagleV-fire with robotic-cape.
- Create I2C device tree to test barometer on robotic-cape.
- Create merge request for I2C support.
- Discuss results and features with mentor.

3.2.12 Milestone #9 (Aug 5th)

- Test all pre-work for librobotcontrol and robotic-cape with beaeglV-fire.
- Upgrade robotic_cape.dts file gateware for beaeglV-fire using pre-work.
- Create Documentation and feeback from mentors.

3.2.13 Milestone #10 (Aug 12th)

- Finalize the work on robotic-cape.dts for beaeglV-fire and test examples of librobotcontrol.
- Create documentation for current process.
- Fixing other bugs, typos, etc. found during documentation.

3.2.14 Final YouTube video (Aug 19th)

- Submit final project video, submit final work to GSoC site and complete final mentor evaluation.

3.2.15 Final Submission (Aug 24th)

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)

3.2.16 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 4

Experience and approch

Experience:

- I'm well experienced with Embedded System and C. I've in-hand experienced with Embedded programming and Hardware design for various boards and projects.
- Here are my projects which demonstrate my proficiency in Embedded system and Robotics.

1. **Martian rover used in IRC (International rover challenge)**
    - Martian rover is a prototype of curosity the nasa mars rover which performed function like soil testing, sample collection and monitoring planet.
    - Project required Embedded hardware and firmware design for motor control, arm control and science sensor's configuration with ROS.
2. **STM32 custom board**
    - STM32 was custom boad which is made in purpose to learn Embedded programming and hardware design. it's a open source development project.
3. **Vaayu – AQI and various concentration calculation for gases present in air**
    - VAAYU is air quality monitoring system device which calibrate the different gases concentration and display with a GUI and TFT-display.
4. **TVC rocketry – Thrust vector control**
    - TVC rocketry is learning based model project about Thrust vector control rockets, which based on PID implementation and sensors configuration.

More projects done by me can be found on my github.com/Himanshukohale22. I've designed various double and four layer board for clients and projects using Kicad , Eagle and Altium designer (Designs). And this shows that I've very good understanding for reading schematics and Circuit design for embedded development, which is required for This project.

Approach:

In my experience, projects often demand a comprehensive understanding of both software and hardware components Before changing the main packages, Hardware setup and debug will required more time than software. This involves meticulous reading of documentation and references, demanding patience and focus. I believe that this content can be completed without any problems.

4.1 Contingency

What will you do if you get stuck on your project and your mentor isn't around?

Unexpected software and hardware problems are most common in any projects. In such cases,

1. In the event of encountering compatibility issues between BeagleBoard and librobotcontrol, I'll to use the BeagleBone Black (BBB) platform for testing purposes, as BBB offers native support for the librobotcontrol package.
2. If there is any hardware related issue to board,first ill review the datasheets and manule of hardware and if there is any issue related to circuitry I'll use oscilloscope, multimeter and other testing devices for debugging.
3. If the problem is about SOC, I'll check the datasheets of perticular SOC.
4. **Here are a few references you can quickly glance at during debugging for guidance.**
    - librobotcontrol package Documentation
    - librobotcontrol github
    - Getting started with beaglebone AI-64
    - Getting started with beagleV-fire
    - Device tree: github.com/Himanshukohale22/BeagleBoard-DeviceTrees , example blog , FDT , ref tutorial
    - Cape interface docs
    - TDA4VM device tree
    - Validatin scripts for understand device tree

4.2 Benefit

If successfully completed, what will its impact be on the BeagleBoard.org community? Include quotes from BeagleBoard.org. community members who can be found on our Discord and BeagleBoard.org forum.

- Librobotcontrol packages will support the beaglebone-AI, beaglebone-AI 64 and BeagleV-fire.
- Various tutorials, Documentation will be added to the Robotic Capes to help the user understand how to use it using llibrobotcotrol packages.

4.3 Misc

Please complete the requirements listed in the General Requirements . Provide link to merge request.

- **All prerequisite tasks have been completed.**
    - Source dive for Librobotcontrol packages and read all the documentation for packages
    - Check hardware specification, setup and device trees for BBB.
    - Here the 'Hello world' cross-compilation task Pull request : merge request