


Drone Cape for BeagleV-Fire - Saish Karole


BeagleBoard.org
Apr 17, 2024

Table of contents

Chapter 1

Introduction

PolarFire SoC platform will allow to create a powerful Drone cape with multiple motors control (octocopter) in FPGA fabric, avionics / flight control on one of Risc-V cores and image processing/ML on FPGA fabric

1.1 Summary links

- **Contributor:** Saish Karole
- **Mentors:** Jason Kridner
- **GSoC:** Proposal Request

1.2 Status

This project is currently just a proposal.

1.3 Proposal

- Completed all the requirements listed on the ideas page.
- The PR request for cross-compilation task

1.4 About

- **Forum:** u/saish_karole (Saish Karole)
- **OpenBeagle:** NachtSpyder04 (Saish Karole)
- **Github:** NachtSpyder04 (Saish Karole)
- **School:** Veermata Jijabai Technological Institute (VJTI), Mumbai
- **Country:** India
- **Primary language:** English, Hindi, Marathi
- **Typical work hours:** 10AM-7PM Indian Standard Time
- **Previous GSoC participation:** This is my first time participating in GSoC.

Chapter 2

Project

Project name: Drone Cape for BeagleV-Fire

2.1 Description

Overview

Drones, an aerial vehicle that receives remote commands from a pilot or relies on software on autonomous flight. This unmanned vehicle, although sounds easy to build, but in reality is a difficult bird to soar. A Drone Cape designed for BeagleV-Fire, by using its PolarFire SOC platform and on-board FPGA fabric to generate a stable PWM for multiple motors across the Drone to help it acheive a stable flight and can land and take off easily. Image Processing algorithms can be used to process Images on FPGA fabric for obstracle detection during mid-air flight.

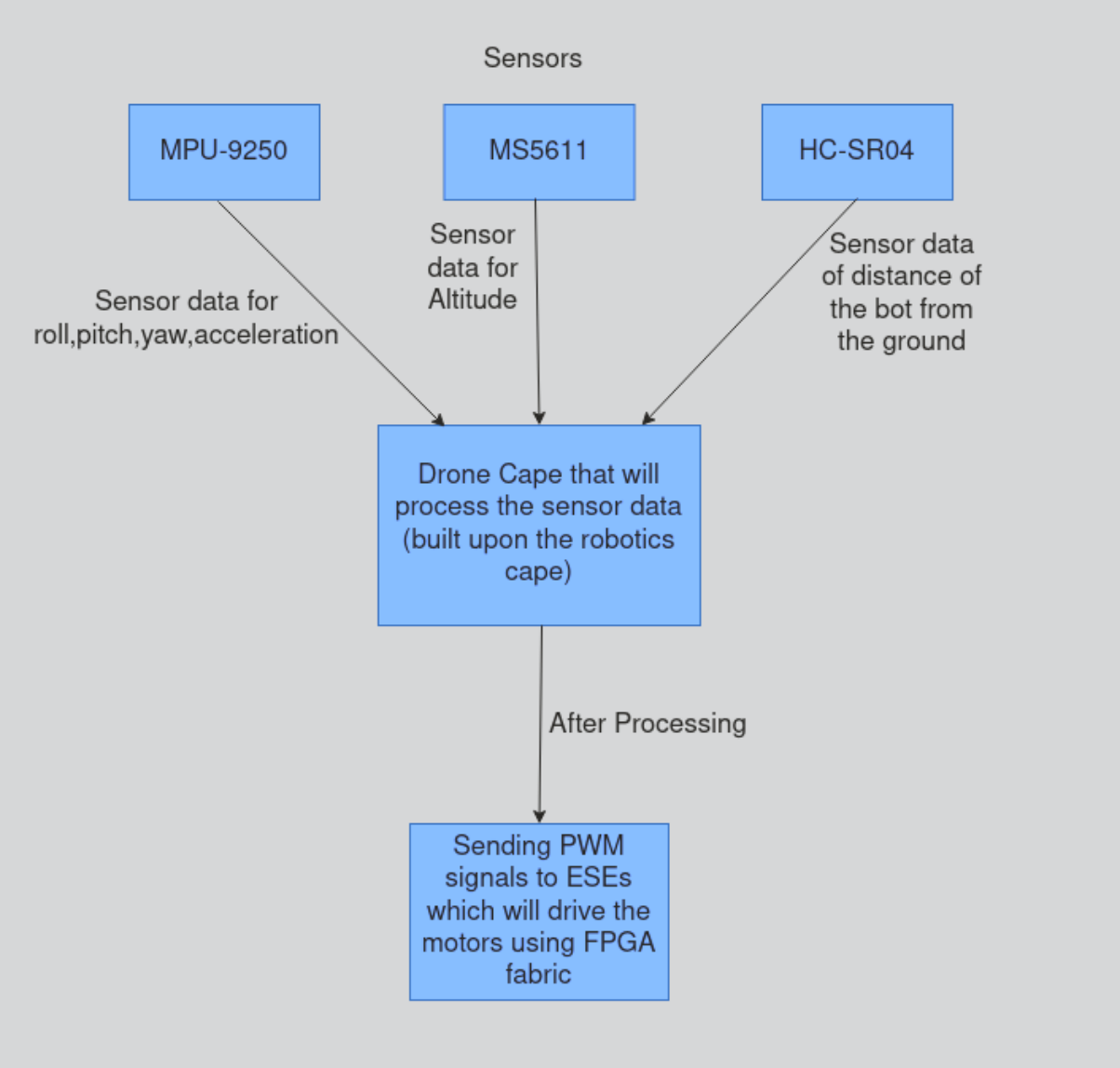
This will be acheived by following technologies:-

- Verilog - Verilog HDL, as the name suggest, is an Hardware Description Language that is used to describe digital systems and circuits in the form of code. It is widely used for design and verification of application-specific Integrated circuits (ASICs) and field-programmable gate arrays (FPGAs) and supports a range of level of abstraction, from structural to behavioral, and is used for both simulation-based design and synthesis based design
- Verilator (For Verification)- Verilator is a free and open-source software tool which converts Verilog (a hardware description language) to a cycle-accurate behavioral model in C++ or SystemC. The generated models are cycle-accurate and 2-state; as a consequence, the models typically offer higher performance than the more widely used event-driven simulators, which can model behavior within the clock cycle. We can test multiple test cases for specified HDL file which will verify the given file
- Microchip FPGA tools- The SoftConsole and Libero tools from Microchip are required by the bitstream builder. Instances of Libero are on git.beagleboard.org's gitlab-runners.
- C++ - C++ will be used to generate scripts of acquring images for image processing and generatng test cases for verification of RTL design

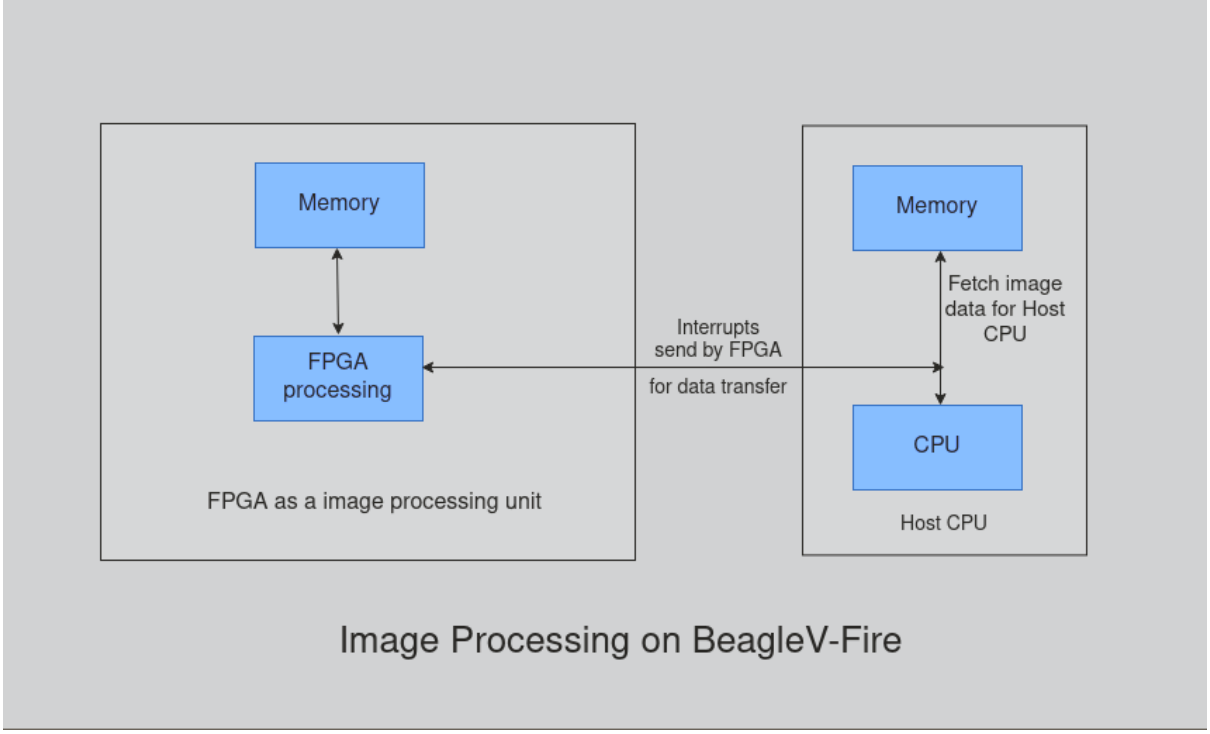
Workflow

- BeagleV-Fire currently has support for robotics cape, Taking inspiration from current robotics cape firmware, I will design a drone cape to handle sensor values for the drone and send PWM to the motors and perform Image Processing on the FPGA fabric.
- PWM generation for motors will be done by writing RTL logic to fetch sensor data for altitude, acceleration and position in x, y and z coordinates in the plane and send PWM to motors according to this data, in Verilog and verify it with the help of Verilator for multiple test cases as well as edge cases and false cases

- Once every RTL file is verified individually, An combined verification i.e verifying the whole rtl kernel of PWM generation after integrating every verilog file, will be performed to verify if all the RTL files are working together without any discrepancies and give correct and expected outputs.
- The drone will be equipped with MPU9250, MS5611, and HC-SR04 sensors. Integration of these sensors is possible with communication protocols like as SPI and I2C.



- For Image Processing, instead of using a camera module and get real time images, I will be using pre-captured images and process it on FPGA Fabric, performing operations like convolutions and edge detectiom
- This will done by converting the given image in greyscale and pixels of greyscaled image into which will be sent to FPGA fabric according to interrupts sent by FPGA to the host CPU. This acquistion of acquring image from Host CPU's memory to the CPU will be performed by C++ script.
- The image will be convoluted with 3x3 predefined kernel in the RTL design file and the resultant pixels values will be send back to on-board RISC-V CPU and will be displayed by the CPU.



2.2 Software

- Verilog
- Verilator
- Microchip FPGA tools
- C++
- Linux
- I2C and SPI
- OpenBeagle CI

2.3 Hardware

- BeagleV-Fire
- MPU9250 IMU board
- MS5611 barometer board
- HC-SR04 Ultrasonic sensor

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

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

GSoC contributors get to know mentors, read documentation, get up to speed to begin working on their projects

3.4 Coding begins (May 27th)

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

- Due to my End Semester examination being conducted in this timeframe, I won't be able to contribute much and hence, a small milestone for the week. Thank you for understanding.
- Setting up remote access/ssh on the BeagleV-Fire and flash led-blink code by customizing the cape gateware by following given documentation. .

3.6 Milestone #2 (June 10th)

- Due to my End Semester examination being conducted in this timeframe, I won't be able to contribute much and hence, a small milestone for the week. Thank you for understanding.
- Analyse the robotics cape for BeagleV-Fire and have preliminary structure of RTL designs for generating PWM ready, written in Verilog.
- These RTL designs will be of fetching the sensor inputs for altitude, acceleration and position of the drone, and generating PWM to be in stable position

3.7 Milestone #3 (June 17th)

- Completing RTL designs for generating PWM and start Verification of individual designs following documentation given on Verilator
- Verification is done through Verilator by writing a C++ script and giving it test inputs to analyse the outputs
- The results of the verification can easily be viewed as waveforms through simulation softwares like GTKWave

3.8 Milestone #4 (June 24th)

- Complete individual and combined verification of the all RTL designs of PWM generator.
- Integrating sensors with the board.

3.9 Milestone #5 (July 1st)

- Integrating all the designs with the drone cape and flash it on the board with the help of OpenBeagle CI, with all the sensors connected to the board.
- Work on the feedback received by the mentor.

3.10 Submit midterm evaluations (July 8th)

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

3.11 Milestone #6 (July 15th)

- Writing an acquistion script for host CPU in C++ and establish a way of transferring data from RISC-V linux to FPGA fabric and vice-versa.

3.12 Milestone #7 (July 22nd)

- Have a preliminary structure of RTL designs for image processing ready and verify whether required operations are being performed or not.

3.13 Milestone #8 (July 29th)

- Integrating RTL designs of Image procesing with acquistion scripts and verify whether image processing has been performed correctly from uploading the image till we get a transformed image as a output by flashing the JTAG on the board.

3.14 Milestone #9 (Aug 5th)

- Adding some example images for the image processing and test both drone cape and image procesing algorithms simultaneously on the board.

3.15 Milestone #10 (Aug 12th)

- Work on the feedback received by the mentor
- Organise the code repository and start working on final project video and report

3.16 Final YouTube video (Aug 19th)

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

3.17 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)

3.18 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

This project requires knowledge about Verilog, FPGA programming and Verification

- I have previously worked on a RISC-V-CPU project , and have a good idea about Verilog syntax and RTL designing.
- In the E-yantra robotics competition, in which I also took part, a RISC-V CPU was developed to find the shortest way through a maze between two points. As a result, I know how to use Verilog to produce PWM and operate motors as well as how to obtain sensor readings through the board's GPIOs and convert them through an ADC for later use.
- I am passionate Open Source enthusiast and I will do the work wholeheartedly. I have my commitment to GSoC and I would do everything in my power to finish the project idea within the allotted time.
- I will keep contributing to the project after GSoC and will be interacting with the community often.

4.1 Contingency

If I get through any contingencies, I will refer the following resources: - I will find resources that are available online. So if I get stuck I will refer those resources. - I will use Beagle forum to communicate with the community to clear my confusion

4.2 Benefit

- A Drone Cape will make building drones easy for drone enthusiasts who don't want to dive deep into making drones from scratch but want to personalize it for their use.
- Similiarly, A drone cape that can be easily mounted on the board and ready to fly will be extremely helpful for rescue or search operations where human intervention is not possible and for agricultural purposes like spraying fertilizers on the crops over acres of land in short amount of time
- A lot of businesses have expressed interest in drone technology because they believe they will save labour expenses for delivering items from one point to another

4.3 Misc

- The PR request for cross-compilation task
- Requested for a free Libero license for building the bitstream locally instead of using OpenBeagle CI