International Journal of Scientific & Engineering Research Volume 12, Issue 3, March-2021
ISSN 2229-5518

RISC-V ISA based Autonomous Quad-Plane using Shakti C-64 Vajra Processor

Shrihari G, Lakshana K, Nikhil Shanmugam A, Radhakrishnan K R

Abstract— Unmanned Aerial Vehicles have become increasingly popular in a plethora of disciplines ranging from cinematography, transportation, healthcare, and critical applications like defense, search and rescue etc. The security, customizability and the reliability of UAVs for such applications play a vital role. This emphasizes the adoption of an open-source royalty free Instruction Set Architecture, RISC-V. This allows for maximum scalability and flexibility in modifying the instruction set according to the requirements, contrary to the proprietary instruction set by ARM. Shakti microprocessors, an open-source microprocessor initiative by RISE Labs, IIT Madras solves this issue by providing an open-source processor cores which can be customized accordingly to suit the needs of the developer [1]. The project aims at designing and developing an indigenous multipurpose, fully customizable flight controller using Vajra C64 microprocessor realized in Arty 7-100T FPGA board [2]. This flight controller design could be easily adopted to other RISC-V based microprocessors and the control system design is also made to be processor independent by the use of SIMULINK Aerospace Block set with Flight Code generated from the SIMULINK coder. The developed control system and the proposed fight controller would prove to be an effective solution against common security flaws, like backdoor exploits in proprietary processors. Fixed wing drone aircrafts require long runways for take-offs and landing. Whereas multi-copters like quadcopters, hexa-copters do not require a long runway. But such multirotor drones are not suitable for long distance quick flight which are of prime importance in defense applications. To solve this issue, a multirotor quad-plane which functions as a quadcopter during take-off and landing, and as an aircraft during flight is proposed. The integration of quadcopters and fixed wing airplanes allows for unique and strategic maneuvers. The indigenously developed flight controller emphasizes the "Make in India" campaign and the "Aatmanirbar Bharat" initiative

Index Terms— Embedded C Code, Fight Control System, Flight Controller, Quad-Plane, RISC-V, Shakti Microprocessor, SIMULINK Model, Unmanned Aerial Vehicle

1 INTRODUCTION

The usage of Flight Controllers plays a vital part in developing autonomous and semi-autonomous aerial vehicles, which are used in a variety of domestic, industrial and military applications. The flight controller can be used for varied purposes ranging from an indigenously developed innovative domestic prototype to a commercial product. This covers the drones used in remote areas to deliver packages, medical related transportation to inaccessible areas, military surveillance, disaster mitigation and farming. A drone with vertical takeoff and landing (VTOL) capability eliminates the need for a runway, hand launch, or catapult for takeoff, and helps ensure soft landings that protect payloads, the operator, and the drone from damage [3]. The development of indigenous Multipurpose Quad Planes allows for quicker deployment into critical sectors like defense, healthcare and other civilian purposes with higher efficiency and reduced costs. The use of Shakti C- Class (Vajra C-64) can be extended to perform on board computations, autonomous navigation, Machine Learning etc., in the future after the development of the basic firmware [4]. To enhance on board computations a companion computer (such as raspberry pi) is added to support pre-built libraries currently. This solves the connectivity problems and the need of a receiving station is eliminated in the cases where the Unmanned Aerial Vehicle is required to navigate to remote locations.

2 METHODOLOGY

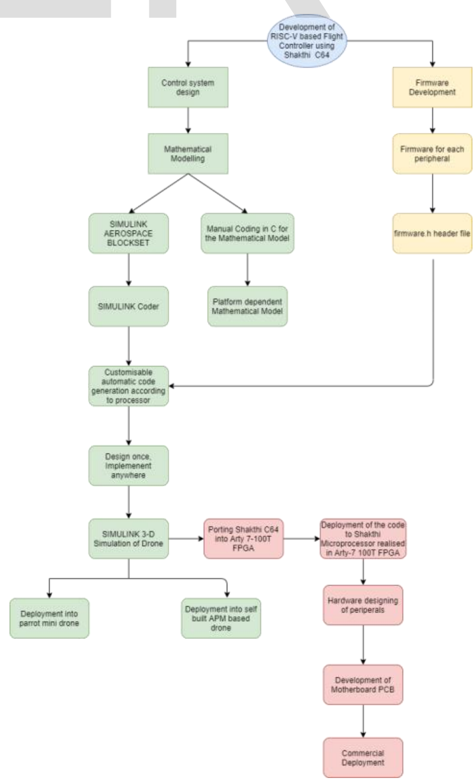

Fig 2.1 Block diagram of the development workflow

3 CONTROL SYSTEM DESIGN

A mathematical model was developed for a Quad-rotor Drone at first and modelled using SIMULINK Aerospace Block set. The control systems also simulate the external environment and receives feedback via the sensors forming a closed loop and controls the actuators to maintain the stability of the system at test. The development of control systems can be done in two ways.

a. Developing and modelling the system natively in Embedded C.

b. Developing the model in Simulink Aerospace Block set and generating the flight code with SIMULINK Coder.

By designing the control systems using SIMULINK it can easily be scaled, modified and improved. This allows the flight control code to be processor independent facilitating the maximum portability of the code. The objective of adopting the SIMULINK based workflow emphasizes on "Design Once, Deploy anywhere". The flight controller is designed with up to 3 Layers of abstraction.

3.1 Quadcopter Model

Quadcopter model **is** the top most layer of the design. The flight control system which is the main part of the control system is considered as a sub system and the quadcopter is modelled. The RF Signal to the drone is modelled via the signal editor. The four sensors are interfaced with the Flight Control System. The Airframe is considered to be non - linear and is modelled via the Airframe Block.

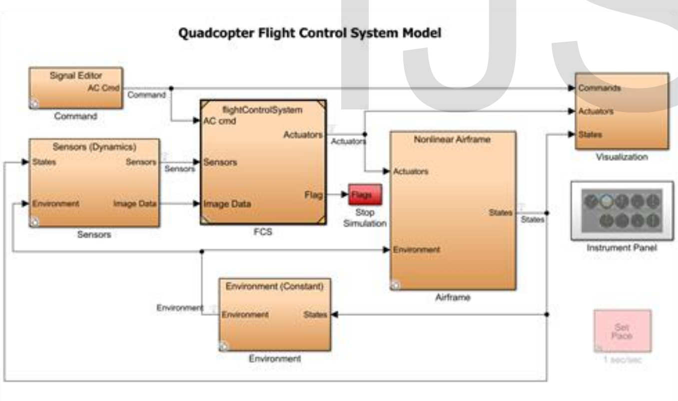

Fig 3.1 Quadcopter Flight Control System model

TABLE 1
Sensors used in the Flight Control System Model

S.No	*Sensor*	*Uses*
1.	Ultrasonic	It is used to measure the distance from the immediate obstacle by sending an ultrasonic pulse and measuring the time taken by the pulse to return back
2.	Pressure	It is used to measure the altitude indirectly using the inverse relation between the pressure and altitude as the ultrasonic waves cannot travel larger distances
3.	Inertial Measurement Unit	This consists of a gyroscope and an accelerometer. The three-axis gyroscope measures the orientation of the drone and the accelerometer measures the acceleration.
4.	Image (Camera)	The onboard camera is used to perform image processing and enable autonomous navigational capabilities. Here the camera uses a technique called as optical flow to estimate the horizontal velocity of the drone by comparing two images taken at a specified interval

A. Flight Control System

The Implemented Flight system consists of the data from the sensors, camera and the RF Signal from the remote controller as the input and sends the actuating signals to the motors and an emergency flag as outputs. The Flight Control System also includes an Image Processing System Block and this implements the optical flow algorithm to estimate the instantaneous velocity of the drone.

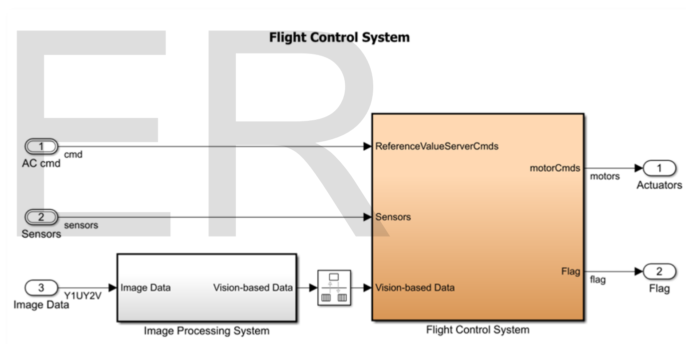

Fig 3.2 Flight Control System SIMULINK block

State estimator performs the core mathematical modelling using the various data obtained from the sensors. The State Estimator block is the used to interpret the angular orientation of the system from the Inertial Measurement Unit, acceleration from the accelerometer, battery level from the battery level indicator, altitude from the ultrasonic and the pressure sensors.

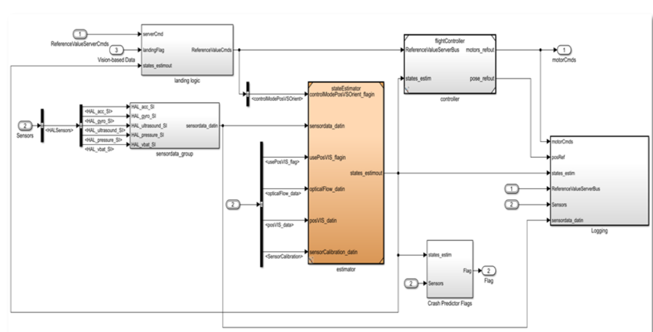

Fig 3.3 State Estimator Design in SIMULINK

B. Thrust Control System

The thrust control system interprets the position of the drone in the drone reference frame mapping from the real-world reference frame. The Position Controller also employs PID Controller to control the roll and pitch. There are six degrees of freedom composed of three translational and three rotational degrees of freedom.

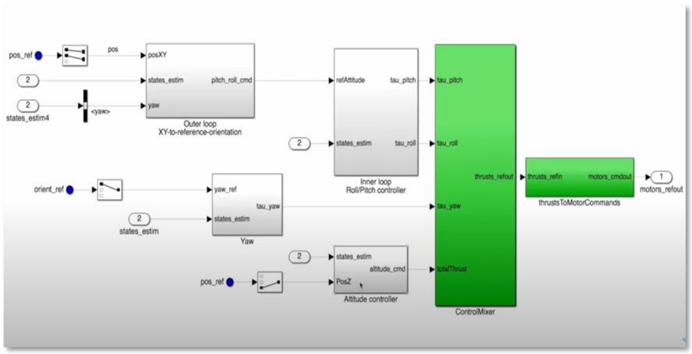

Fig 3.4 Thrust Control System Design in SIMULINK

4 Code Generation - Demonstration

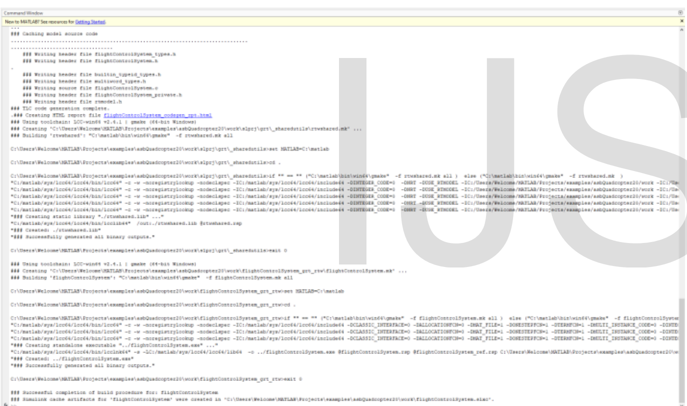

Fig 4.1 Generation of C code from SIMULINK coder

The generated code is then dumped and tested in Parrot Mini Drone and APM based Flight controller. The workflow of adopting automatic code generation allows for deployment in any processor of choice.

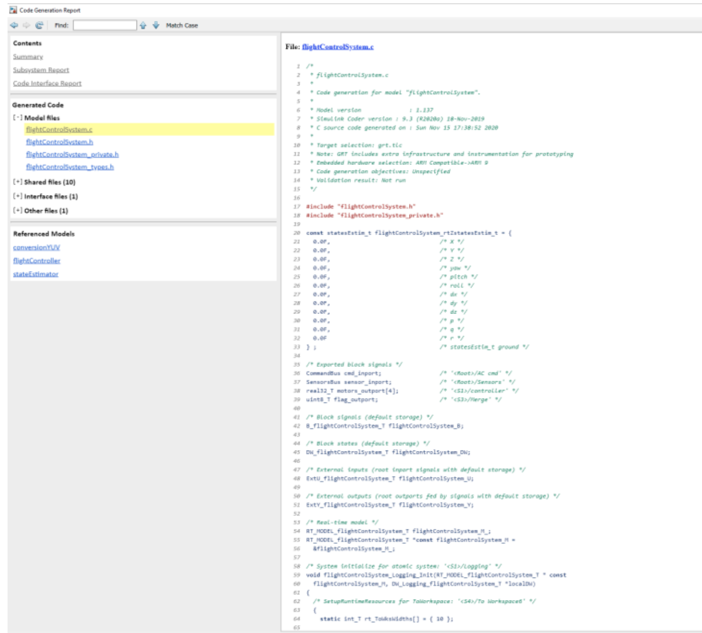

Fig 4.2 Snippet of "flightcontroller.c" – C code generated by SIMULINK coder

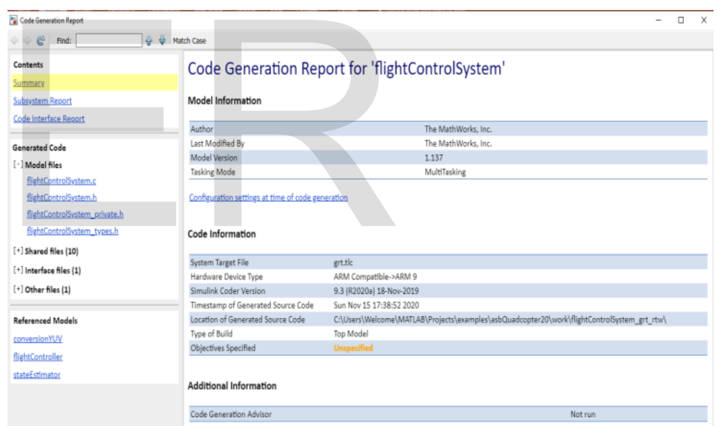

Fig 4.3 Code Generation Report from MATLAB

The control system code can be optimized for the performance by a suitable trade-off between the RAM allocation and the time for instruction execution. This makes the code less RAM intensive as the Shakti Vajra C64 ported in Arty 7-100T comes with only 256 MB of RAM. This also proves that a clock frequency of more than just 50 MHz suffice to fulfill the application without much troubles [5].

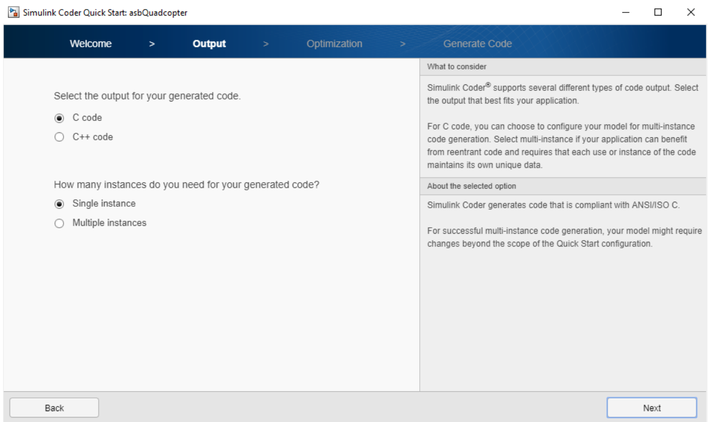

Fig 4.4 Simulink Coder for Quadcopter model

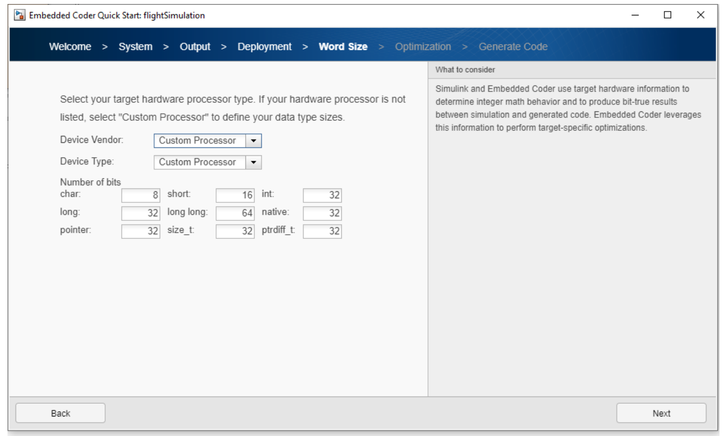

Fig 4.6 Embedded Coder – custom processor (Vajra C64) configuration

The workflow of deploying the drone with control system developed as SIMULINK blocks also allows for the possibility of a tight integration with ASIC level design via automatically generated optimized HDL Code from the MATLAB toolset.

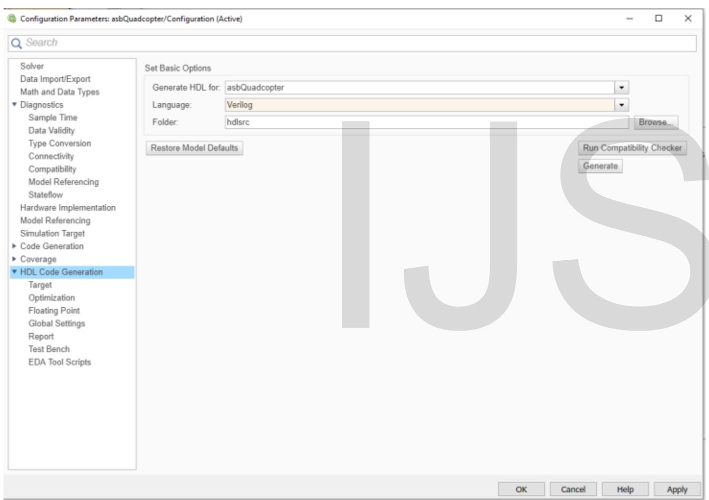

Fig 4.5 HDL coder for the SIMULINK quadcopter model

A. Deployment on Custom Processor

The developed control system can be deployed on any processor making the workflow a design once and deploy everywhere approach. This approach reduces the development time improves scalability and flexibility in adopting the control system for any processor in the future. This will also support RISC-V ISA variants as the data type size can be fixed.

The Control system though developed for RISC-V based Shakti C64 Vajra Microprocessor, the control system has been implemented on APM based drone as well as the commercially available Parrot Mini Drone.

B. Deployment on Shakti Processor

Since we opt for Shakthi C- Class (Vajra C-64) a 64-bit processor and we would have to several peripherals to the board and perform on board computations for autonomous navigation. C-Class SoC has the RV64IMAFDC making it an ideal candidate for on-board DSP and other computations. The E-Class processor does not have Floating point instruction set so it can't be used for on-board Processing. Moreover, the Fault tolerant variants of the C-Class Processor for ISO26262, under which the UAV for military and critical operations comes into, we go for Vajra C64 realized using Arty 7 100 T.

5 Simulation and Flight Plan

The deployed control system is simulated using the Aerospace Block set simulator and Parrot Mini-Add-On in MATLAB. The drone was able to hover at the start of the simulation and was able to maneuver the planned trajectory. The trajectory test succeeded as in Fig. 5.5

The instrument cluster is also designed to output the different flight parameters and telemetry data namely

1. Altitude
2. Airspeed
3. Virtual Heads on Display
4. RPM of the four rotors
5. Magnetometer
6. Vertical Speed

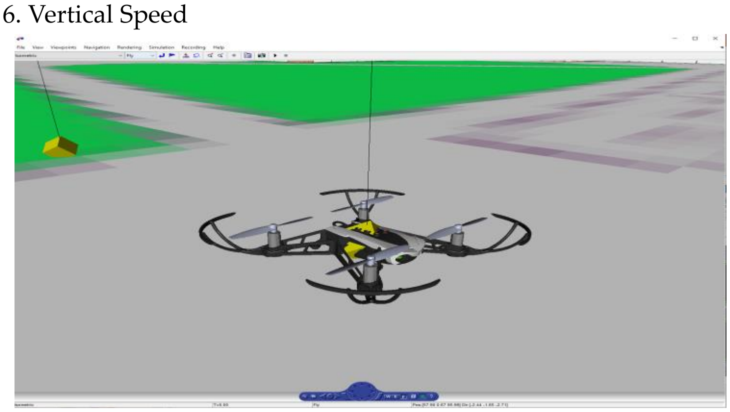

Fig 5.1 Quadcopter simulation using custom developed control system

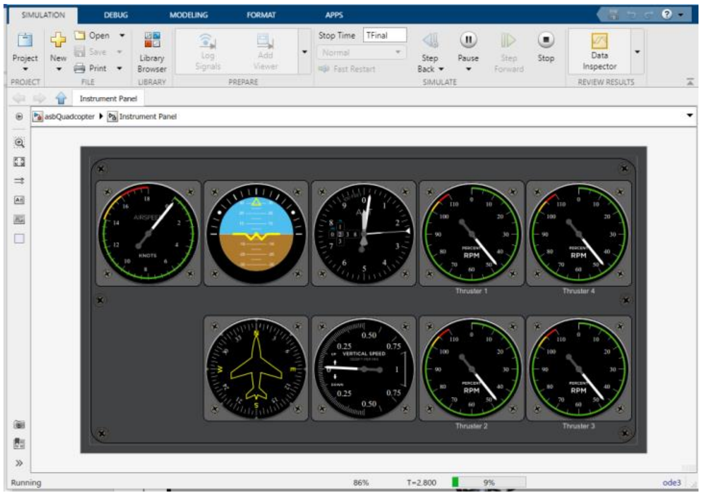

Fig 5.2 Developed Instrument cluster for the quadcopter during the simulation

```
### Successful completion of build procedure for: flightControlSystem
### Simulink cache artifacts for 'flightControlSystem' were created in 'C:\Users\Welcome\MATLAB\Projects\examples\asbQuadcopter20\work\flightControlSystem.slxc'.
Running trajectoryTest
.
Done trajectoryTest
__________

ans =

  TestResult with properties:

          Name: 'trajectoryTest/testTrajectoryGeneration'
        Passed: 1
        Failed: 0
    Incomplete: 0
      Duration: 0.5761
       Details: [1×1 struct]

Totals:
   1 Passed, 0 Failed, 0 Incomplete.
   0.57615 seconds testing time.
```

Fig 5.3 Trajectory test result

In order to dump the code in Vajra C64 processor the .mcs file and bit stream from the GitLab Repository of RISE LAB is cloned and uploaded into the Arty 7-100T FPGA. The PlatformIO IDE detects the board and the generated C code with suitable modifications according to the PINMUX mapping of Vajra C64 is dumped.
The firmware for the peripherals is written. At first I2C is written and the codes are used in PlatformIO IDE to debug and evaluate performance using spiking. The design is verified within the IDE.

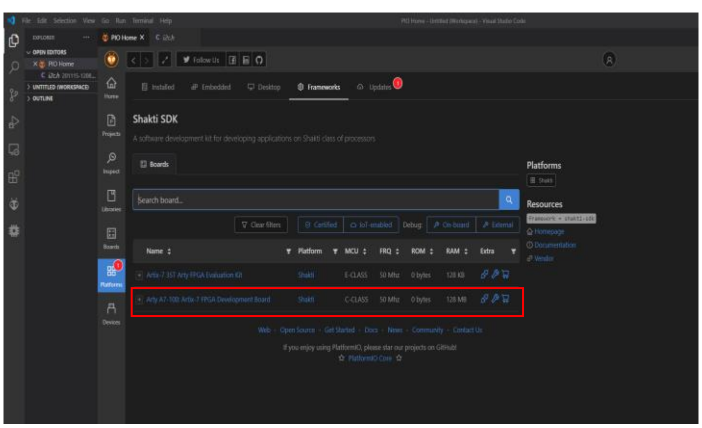

Fig 5.4 Arty 7-100T FPGA board module configuration

Fig 5.5 Test flight of custom built APM based drone with autonomous navigation

Fig 5.6 Test flight of parrot mini drone with control system implemented in Simulink

6 Features

- No dedicated runway required
- Agile and swift compared to multirotor
- Quick and convenient vertical take-off and landing
- Higher payload capacity
- Gliding capability which saves battery in the Quad Plane leading to higher flight time
- Multi-Purpose design suitable for defence and civilian applications

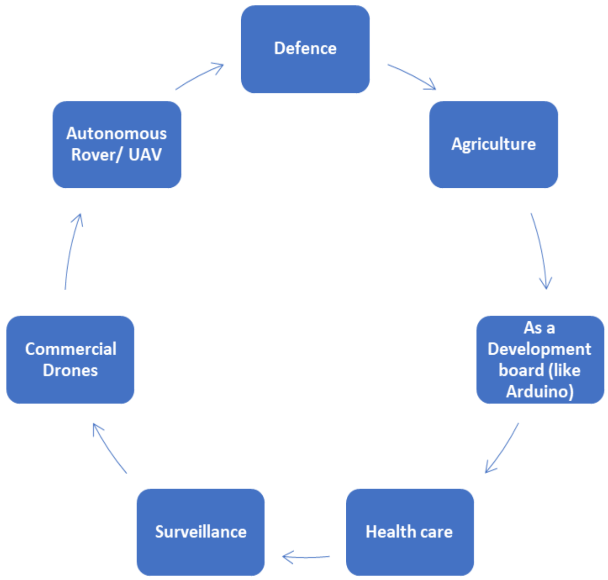

Fig 6.1 Trust areas for Shakti based Quad planes

7 Conclusion

The proposed indigenous autonomous quad plane will be suitable for deployment into military and defence applications providing a secure and backdoor exploit less Controller for Mission Critical Applications. The flexibility of the controller would allow for the deployment for civilian applications ranging from agriculture to transportation emphasizing the "Make in India" campaign and the "Aatmanirbar Bharat" initiative.

Acknowledgment

We are thankful to Mr. S. Sainath, Assistant professor, Department of Electronics and Communication Engineering, PSG College of Technology for his valuable guidance and support. We also thank Mr. K.R Radhakrishnan, Project Guide & Assistant professor, Department of Electronics and Communication Engineering, PSG College of Technology for his immense support.

References

[1] N. Gala, A. Menon, R. Bodduna, G. S. Madhusudan and V. Kamakoti, "SHAKTI Processors: An Open-Source Hardware Initiative," 2016 29th International Conference on VLSI Design and 2016 15th International Conference on Embedded Systems (VLSID), Kolkata, 2016, pp. 7-8.

[2] SHAKTI C class micro architecture design https://gitlab.com/shaktiproject/cores/c-class

[3] Tailsitters vs Quadplanes – Why a VTOL tailsitters is the best surveying drone for mapping missions https://wingtra.com/tailsitters-vs-quadplanes-why-a-vtol-tailsitter-is-the-best-surveying-drone-for-your-mapping-missions/

[4] SHAKTI Processor Program Open-source Processor Development Ecosystem https://shakti.org.in

[5] Design of RISC-V Instruction Set Architecture https://riscv.org/specifications/

- *Shrihari G is currently pursuing bachelors degree program in electronics and communication engineering in PSG College of Technology, India, PH-8754749274. E-mail: shariethernet@gmail.com*
- *Lakshana K is currently pursuing bachelors degree program in electronics and communication engineering in PSG College of Technology, India, PH-8903012422. E-mail: luxkathir@gmail.com*
- *Nikhil Shanmugam A is currently pursuing bachelors degree program in electronics and communication engineering in PSG College of Technology, India, PH-9486095796. E-mail: nikhilshanmugam.ns@gmail.com*
- *Radhakrishnan K R is currently working as an Assistant Professor in the Department electronics and communication engineering in PSG College of Technology, India, PH-9148287338. E-mail: krr.ece@psgtech.ac.in*