The Control & Computing System hardware consists of:

* **Flight Control System**

1. Pixhawk 2.4.8 (ARM Cortex M4, 6-axis gyroscope + accelerometer, barometer, compass, failsafe co-processor)
2. NEO-M8N GPS Module (Accuracy of 2m & 18 Hz update rate)

* **Flight Computing System**

1. Raspberry Pi Model 4 B (4GB RAM)
2. Raspberry Pi Night Vision Camera (5MP, 1080p)

* **Telemetry**

1. 433 MHz 500mW radio telemetry (2.5km range)

  The hardware for the control system is selected after a detailed analysis of all available options in terms of cost (time & money), reliability, and mission limitations. Control and Computing systems, both are put inside the UAV for two reasons:

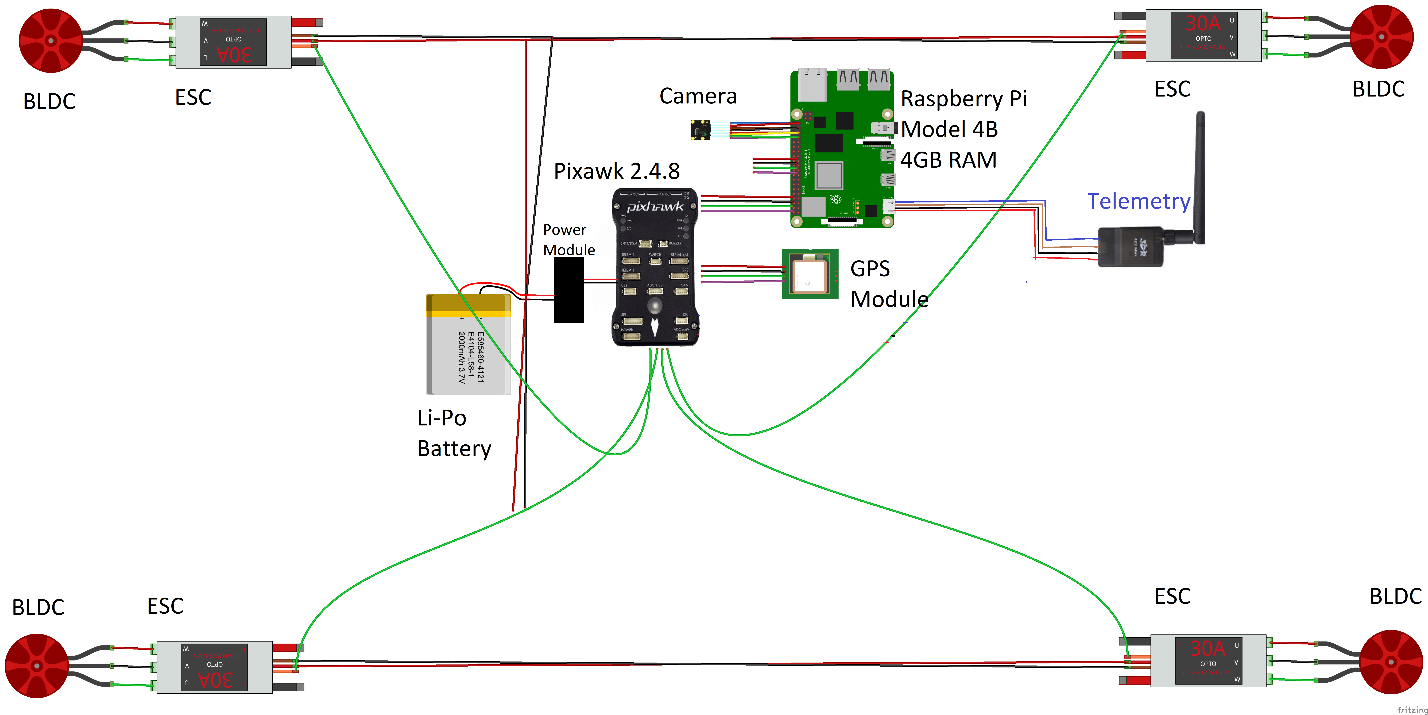
* **Complete autonomy of UAV:**

UAV is independent of any ground control station and can complete spraying missions outside the telemetry range with great precision.

* **Advanced Computing & Artificial Intelligence:**

Using HD image processing, machine learning, advanced model-prediction algorithms & artificial intelligence on a powerful quad-core Raspberry Pi, the onboard computing system makes this UAV the first of its type, which outperforms any manual or autonomous agriculture drone in the market.

The controller and Computer are present on different adjacent boards inside UAV. The schematic below shows the rough idea of electrical connections (not the original electrical schematic, just a rough schematic for better understanding).



The most optimum approach would be to embed the flight controller (ARM Cortex-M4 + 32-bit failsafe co-processor), all control sensors (MPU6000, ST micro gyroscope + magnetometer), flight computer (ARM Cortex-A72, SRAM, DRAM), and all peripherals on a single PCB. It would significantly reduce the wiring across all avionics. The main issue in this approach is the fabrication of this all-in-one PCB. It is estimated to be minimal of 6-layer PCB with <150µm scale, which isn’t available in Pakistan. Moreover, designing all connections, power optimization, signal latency & area optimization require more time and effort. Depending on the time & resources available, this doesn’t seem impossible to achieve soon.

**Flight Controller Software:**

Pixhawk runs modified ArduPilot firmware for quad tilt-rotor UAVs with four rotors tilting. Modifications in the firmware provide optimum performance with our UAV’s aerodynamic specifications. Modifications include:

* Vertical to Horizontal or Horizontal to Vertical Flight Transition with variable tilt-angle (enabling both time & energy optimization according to requirements)
* Drone stabilization by controlling four tilting servos on independent PIDs (for optimum stability)
* Shifting Telemetry to Flight Computer to enhance the performance of control loops in-flight computer since there is only one core in Pixhawk (ARM Cortex-M4)

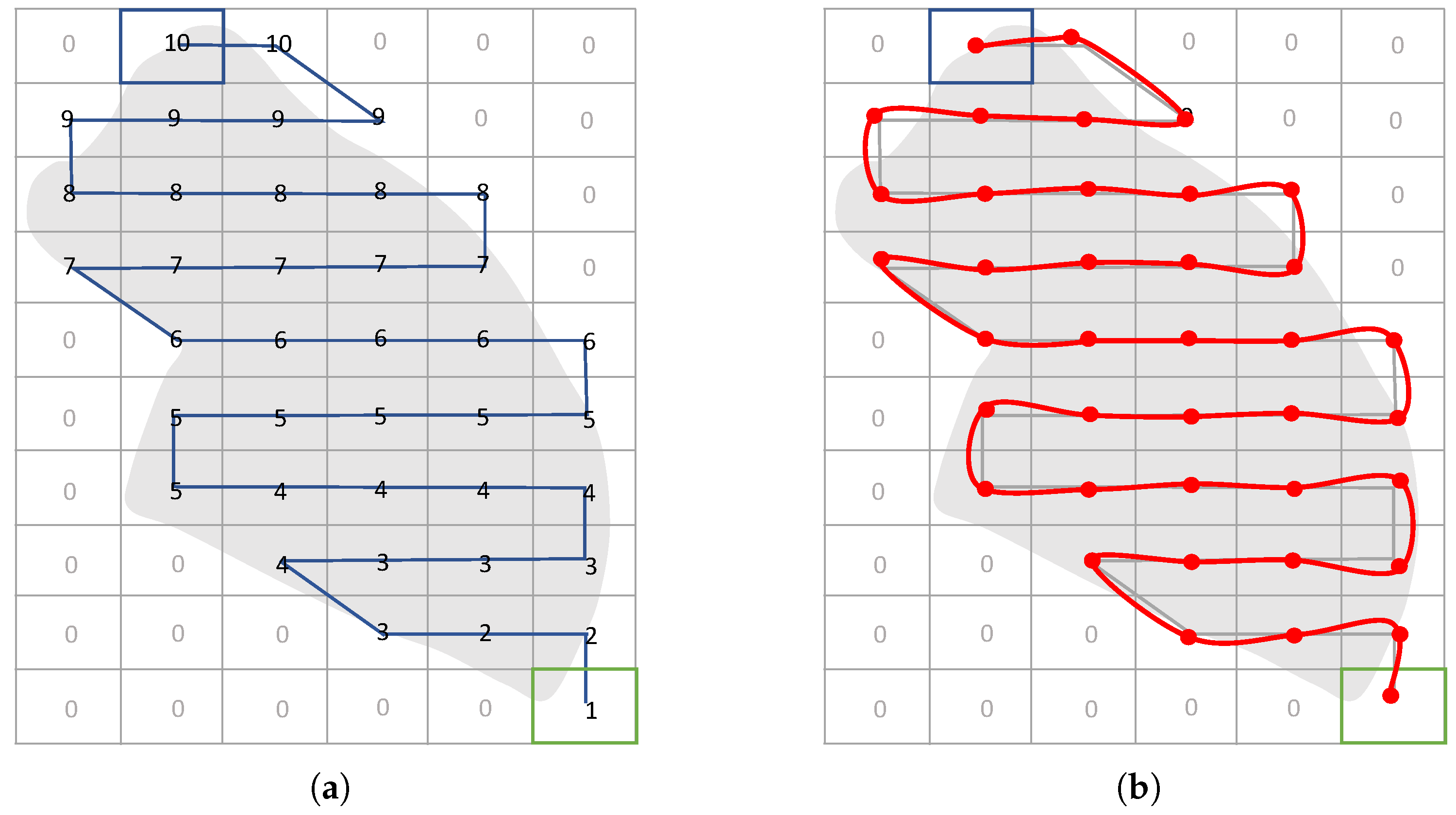
**Image Processing:**

The Raspberry Pi Night Vision Camera enables the drone to capture high-quality images during the day and at night. We use Open-CV written in C++ for faster image processing. It would allow real-time computer vision in the UAV.

In addition to Open-CV, the drone uses with YOLO (You only look once) algorithm for real-time object detection using Machine Learning. It would allow the drone to be fully autonomous since it would be able to recognize objects in the images of the camera and make movements according to the landscape.

**Path Optimization:**

The drone should take the shortest path to complete its task optimally. The drone ensures this with the most efficient path optimization algorithms. The drone would use the Traveling salesman problem algorithm to find the shortest Hamiltonian path.



Now the drone has the shortest path(a), but the drone (in fixed-wing mode) can’t completely follow the path generated by the previous algorithm. It occurs due to the speed of the drone and the maximum turn angle of the drone. For final optimized path, the drone would use the Band-Turn mechanism. It would help the drone to achieve path in the image(b). Thus the drone would use this path to consume the minimum energy needed.