Objective

The objective of Week 3 was to move beyond raw bit extraction and transform the recovered 112-bit ADS-B frames into meaningful, structured aircraft information. At this stage, the project transitions from signal demodulation into protocol-level decoding and validation.

By the end of this week, I aimed to take real ADS-B frames captured from the air, verify their integrity, parse their protocol fields, and decode selected message payloads into human-readable data.

From Demodulated Bits to Valid Frames

In Week 2, I successfully detected ADS-B preambles and demodulated Pulse Position Modulated signals into binary bitstreams. However, demodulated bits alone do not guarantee valid data. RF noise, interference, or imperfect timing can still corrupt frames.

To address this, I implemented a frame parsing stage that strictly follows the ADS-B extended squitter format. Each decoded frame consists of exactly 112 bits with the following structure:

Bits 0–4: Downlink Format (DF)
Bits 5–7: Capability (CA)
Bits 8–31: ICAO Aircraft Address
Bits 32–87: Message Payload (ME)
Bits 88–111: CRC Parity

RC Validation and Frame Integrity

One of the most important steps added in Week 3 was Cyclic Redundancy Check (CRC) handling. Even when a preamble is correctly detected and bits are extracted, a frame may still be invalid due to RF noise or pulse collisions.

CRC provides a mathematical method to verify whether a received ADS-B frame is intact. The receiver recomputes the CRC over the first 88 bits of the message and compares it against the transmitted parity bits. Frames that fail this check are discarded.

Payload Decoding and Message Interpretation

After validating a frame, the next step is decoding the message payload contained in the ME field. The interpretation of this field depends on the ADS-B Type Code, which indicates what kind of information the aircraft is broadcasting.

For this project, I focused on decoding Aircraft Identification messages as a proof of concept. These messages encode the aircraft callsign using a custom six-bit character encoding defined by the ADS-B standard.

Results and Verification

Using this pipeline, I successfully decoded multiple ADS-B frames from real RF captures. The decoded output includes:

- ICAO aircraft addresses
- Downlink format and capability fields
- Callsigns for identification messages
- CRC values for integrity verification