Air Defense Architectural Review

High-Level Block Diagram

External Hardware

- Camera detects scene
- Monitor displays tracked objects
- Motor and laser track and shoot target

FPGA Programmed Units

- GPIO and VGA managers handle external signal routing
- Display image generator takes frame data and position data to produce image
- Image pipeline tracks a thrown object
- Motor pipeline drives the laser to our desired location

FPGA Board Switches

- Handle reset and setting other modes

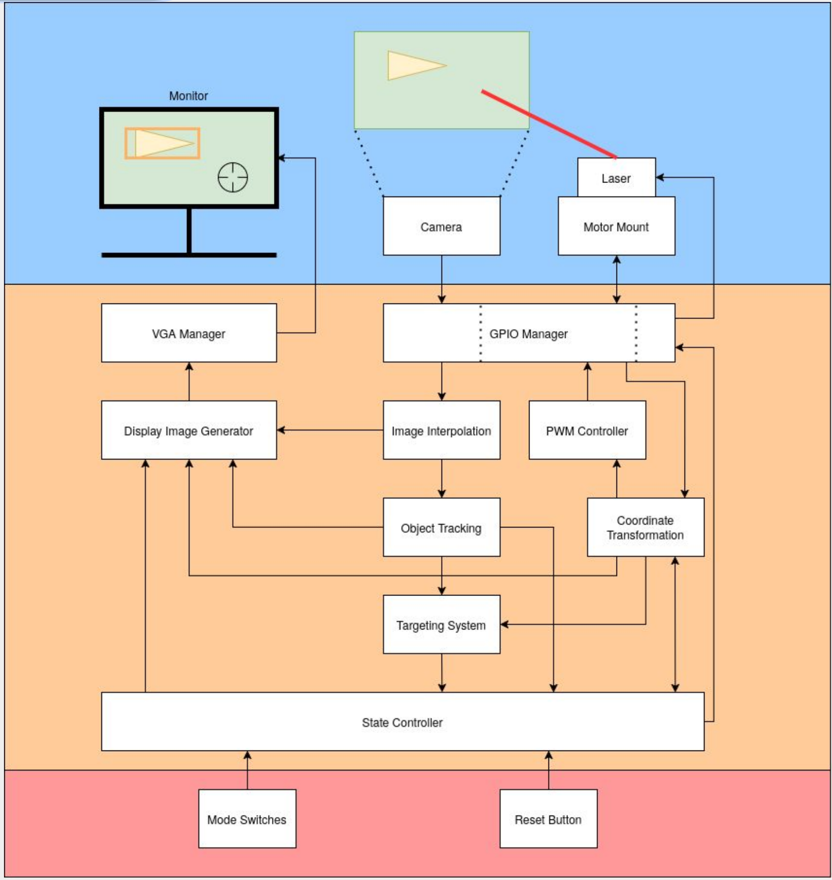


Image Acquisition (Camera)

- 12 bit pixel values, when FVAL and LVAL high
- EXTCLK (96 MHz) ~= PIXCLK
- Active Image is valid data, in bayer pixel format
- Bayer pixel mosaic is a color filter to capture color images with just a single RGB sensor
- G:50% R:25% B:25%
- Each color filter has its own Gain settings to amplify the signal from the image sensor
- Programming for all this available through I2C Block

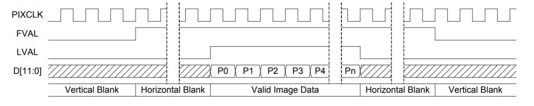
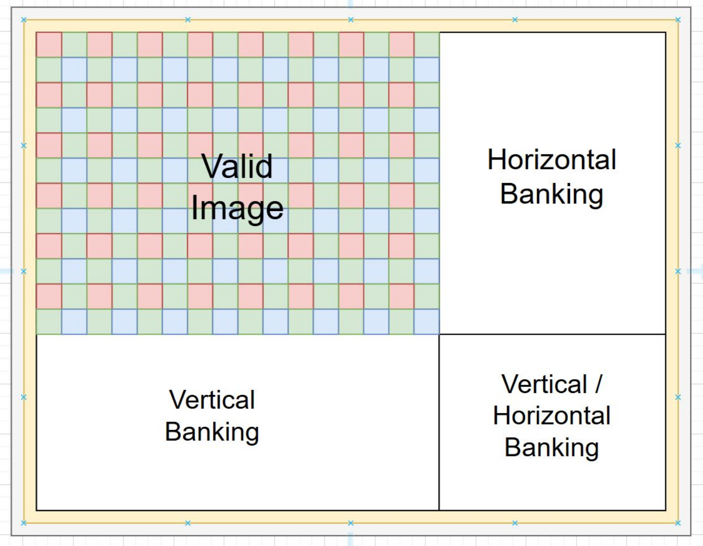


Figure 1.5 Default Pixel Output Timing



Resolution	Frame Rate	Sub-sampling Mode	Column _Size (R0x04)	Row_Size (R0x03)	Shutter_ Width_ Lower (R0x09)	Row_Bin (R0x22 [5:4])	Row_Skip (R0x22 [2:0])	Column _Bin (R0x23 [5:4])	Column _Skip (R0x23 [2:0])
640 x 480 VGA	150	N/A	639	479	<479	0	0	0	0
	150	skipping	2559	1919		0	3	0	3
	77.4	binning	2559	1919		3	3	3	3

Object Tracking System

Training YOLO:

- Recorded videos of a paper airplane and spliced it into frames
    - Ended up with 123 images
- Initially trained YOLO on those 123 images with 25 epochs
- Next Steps:
    - Take more videos in different areas to train YOLO against for validation
    - Output weights and biases for use in the lower-level modules
    - Play around with epoch settings (want to avoid overfitting)

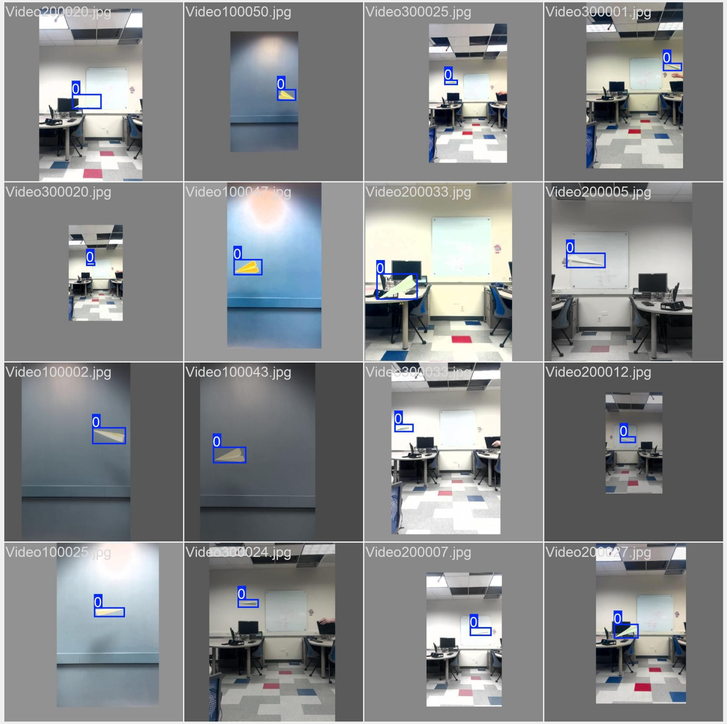
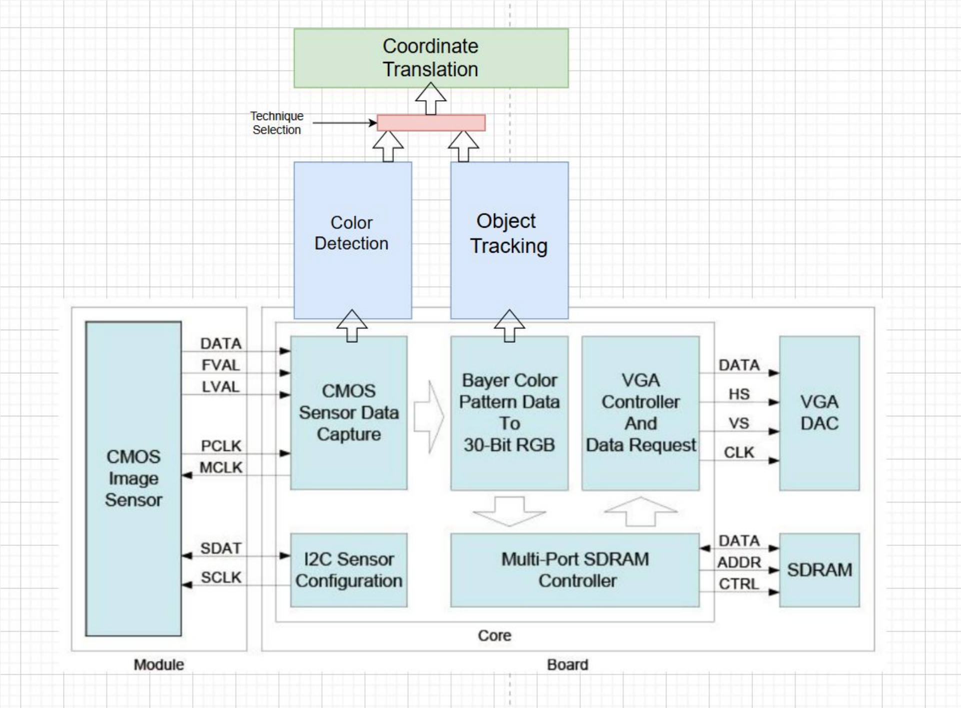
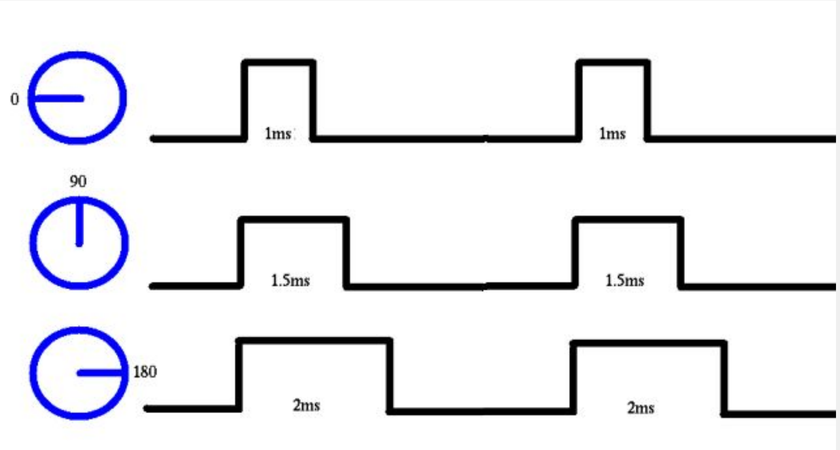


Image Acquisition Data Flow



Servo Motor Control

- PWM Control module will generate square waveform with necessary duty cycle to move the servo to desired position
- Arduino guided testing suggests that the fewer number of times we update the servo position, the smoother the movement
    - Each updated position requires the remainder of the 20ms waveform to be idle for adjustments (internal to servo motor)
- Leads to believe we will require some form of object leading for accurate laser placement considering our estimates 200ms of flight time



Servo Drivers

Our lives will be made easier by implementing servo drivers (similar to arduino's) for interfacing with the servo motors.

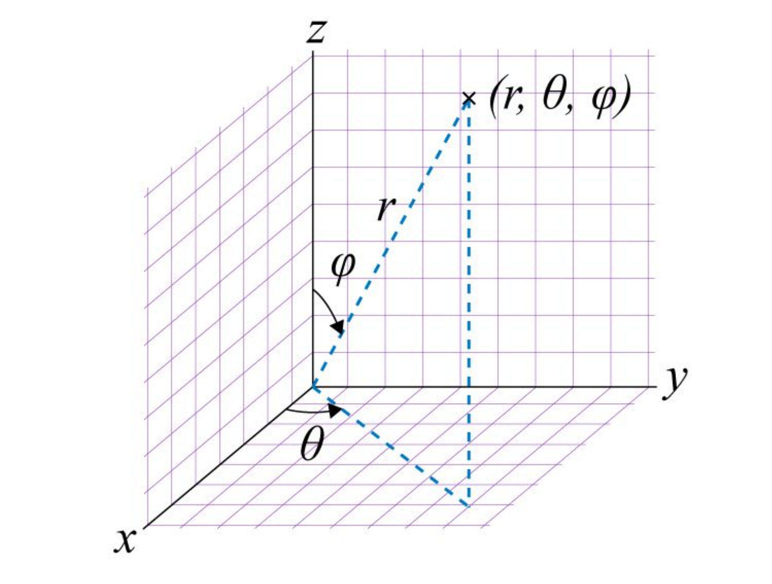
Standardizing the command for servo position will make sure that no matter how we are sending commands for servo movement, it will be the same

Implementation may use a lot of onboard memory. Lookup tables tying duty cycle to degree position

Arduino's open source servo drivers may give good guidance, although principle of generating waveforms is rather straight forward.

Coordinate Transform

- Needs to convert Cartesian coordinates to spherical coordinates for the servo motors
  - Use fixed point numbers
- LUT
  - Fast response but high overhead in setup and memory
- Real-Time Computation
  - More accuracy but higher processing requirements
  - Trigonometry calculations are expensive
- Likely need predictive tracking to compensate and improve accuracy
  - Relatively low hardware cost
  - Assumes predictive flight pattern



Motor Control Data Flow

Considerations

- Linear path -> perform parallel calculations where possible
- Pipelining -> balance throughput with latency
- Noise / Variability -> may need to apply a filter to ensure smooth motor control and tracking