FPGA_Signal_Control_System – Engineering Cheat Sheet

Real-Time Distance Mapping, Temperature & Fan Control, Motion Sensing, Telemetry, VGA Visualization

---

What am I looking at? Hardware-only sensing, control, and visualization lab on a Nexys A7 FPGA.
What does it mean? Every LED, pixel, and UART byte corresponds to a physical quantity in the room.

1. System at a Glance

Hardware platform:

- Digilent Nexys A7-100T FPGA board (100 MHz master clock).
- Pmod ToF (ISL29501), PIR sensor, 5 V fan + MOSFET driver, rotary encoder, VGA, UART.

Major functions:

- Measure **distance** with a ToF sensor and build a 2D range map.
- Measure **temperature** via XADC (and optional TMP I^2C sensor).
- Detect **motion** with a PIR sensor and visualize "recent activity".
- Control a **fan** from temperature, PIR occupancy, and manual inputs.
- Render a **VGA display**: logo viewport, ToF range plot, and HUD.
- Stream structured **telemetry** over UART at 2 Mbit/s to a host PC.

Design philosophy:

- Single 100 MHz clock domain; all slower behavior uses tick-enables.
- Ready/valid-style streaming between modules.
- Fixed-point Q1.15for temperature, angle, and (scaled) distance.
- Clean separation between *sensing*, *control*, *transport*, and *visualization*.

2. What the Board is Doing Physically

2.1 Time-of-Flight (ToF) Sensor – ISL29501

Signals:

- I^2C: `SCL`, `SDA` (open-drain).
- Control: `tof_ss` (sample start), `tof_irq_n` (conversion done, active-low).

FPGA outputs (logic domain):

- `dist_mm`: 16-bit distance in millimetres.
- `tof_status`: sensor status flags.
- `tof_vld`: 1-cycle strobe when a new sample is ready.

What it means on VGA:

- Each `dist_mm` is plotted as a point in the left-half range plot.
- Green pixels = valid ToF hits; gaps = no return / out-of-range.
- Concentric cyan rings mark 1/4, 1/2, 3/4 of the configured max range.

2.2 Rotary Encoder (Survey Spinner)

Inputs: `jc_rot_a`, `jc_rot_b`, `jc_rot_sw`.
Decoded into:

- `rot_step_pulse`: one pulse per detent.
- `rot_step_dir`: step direction (1 = CW, 0 = CCW).
- `rot_pos`: signed 16-bit position counter.
- `rot_btn_pulse`: debounced pushbutton event.

What it means:

- In **AUTO** survey mode: surveyor FSM owns the angle; encoder is a secondary input.
- In **MANUAL** mode: encoder steps directly move the survey angle and HUD spinner.

On the VGA:

- Spinner/compass widgets track `theta_q15`.
- A vertical side bar shows magnitude and direction of motion (green = CW, red = CCW).

2.3 PIR Motion Sensor

Input: `pir_raw` (3.3 V digital out from PIR module).
Conditioned signals:

- `pir_rise`: single-cycle pulse on a new motion event.
- `pir_active`: holds HIGH for a "recent motion" window (e.g., ∼15 s).

Visual cues:

- RGB LED transitions from bright green to off as time-since-motion increases.
- A HUD tile and "motion streak" bar show how recently someone moved.

2.4 Temperature Sensors & Canonical Q1.15Bus

Sources:

- XADC analog temperature or external analog channel.
- Optional digital TMP sensor via I^2C.

Canonical representation:

- `samp_q15`: signed Q1.15temperature code.
- `samp_valid`: strobe for new samples.

Downstream modules do not care which sensor was used; they only see `samp_q15`.
On the VGA:

- Numeric temperature readout.
- Vertical bar with colored zones (cool, comfort, hot).

3. Fan Control & PWM Behavior

3.1 Fan Controller FSM (`temp_fan_ctrl`)

Inputs:

- `samp_q15`, `samp_valid` – canonical temperature.
- `pir_active` – recent motion.
- Manual pushbutton and enable switches (`sw_temp_en`, `sw_manual_en`, `sw_pir_en`).

Outputs:

- Physical: `fan_pwm_ja3` and `fan_en_ja4` to MOSFET + fan.
- Logical: `fan_temp_on`, `fan_manual_on`, `fan_pir_on`.
- `duty_final_q15_dbg`: final fan duty in Q1.15.

Interpretation:

- Temperature hysteresis selects OFF/LOW/MED/HIGH bands.
- PIR and manual paths can override or augment temperature-based commands.

On LEDs:

- Dedicated fan-status LEDs show which input (TEMP/MANUAL/PIR) is contributing.

3.2 PWM DAC (`pwm_dac`)

- Maps any Q1.15input to a 12-bit PWM duty cycle.
- Runs at ≈24.4 kHz (100 MHz / 2^{12}).
- Used both to drive the fan and as a general-purpose analog-style debug output.

4. Surveying, Angle, & Range Map

4.1 Surveyor FSM (`surveyor_fsm`)

Role:

- Controls automatic sweep of the ToF sensor.
- Generates `step_pulse`, `step_dir` and `sweep_wrap`.
- Schedules ToF measurements so each angle gets one distance.

4.2 Unified Step Source

- **AUTO** mode: steps from `surveyor_fsm`.
- **MANUAL** mode: steps from rotary encoder (`rot_step_pulse`, `rot_step_dir`).

Unified signals seen by angle/index and HUD:

- `step_pulse_src`, `step_dir_src`.

4.3 Angle Indexer (`angle_indexer`)

- Maintains `angle_idx`, `theta_q15`, and `theta_turn_q15`.
- One full turn corresponds to `theta_q15` $\in [0, 1)$ (Q1.15).
- A constant `Q15_PER_STEP` encodes angle increment per survey step.

Conversion on PC:

$$\theta_{\text{rad}} = 2\pi \cdot \frac{\text{theta\_q15}}{2^{15}}, \quad \theta_{\text{deg}} = 360^\circ \cdot \frac{\text{theta\_q15}}{2^{15}}.$$

4.4 ToF Framebuffer Writer

Module: `tof_plot_point_writer`

- Converts polar (r = dist_mm, θ = theta_q15) to framebuffer (x, y).
- Writes to a **256×256, 1-bit** dual-port BRAM.
- `tof_frame_clear_fsm` clears the framebuffer on each `sweep_wrap`.

What you see: the left-half of the VGA display is a live 2D map of measured distances.

5. VGA Pipeline: What You See on the Screen

5.1 Timing Core (`vga_range_plot_top`)

- Derives 25 MHz pixel clock from 100 MHz.

- Generates standard 640×480@60 Hz timing.
- Provides `hcount`, `vcount`, `active_video`, `frame_tick`.

5.2 Top-Left Logo Viewport

Region:

- Screen: 160×120 pixels at top-left.
- Logical image: 320×240 RGB444 (`image_dualbuf_320x240_rgb12`).

Behavior:

- Two logo banks (double-buffered BRAM).
- PC streams new images into the *inactive* bank via UART.
- Swap request `logo_swap_req_sys` flips banks on `frame_tick`.
- Slide switch can override which bank is displayed.

Meaning:

- This region is fully user-programmable (logos, heat maps, etc.).
- A small indicator tile can show which bank is active.

5.3 ToF Range Plot (Left Half)

Region:

- `hcount` $\in [0, 319]$, `vcount` in a 256-line window under the logo.

Color code:

- Bit set in framebuffer ⇒ bright green pixel.
- Bit clear ⇒ dim gray background.
- Cyan rings: calibrated radius levels.
- White vertical/horizontal lines: axes and zero-distance baseline.

5.4 HUD Status Overlay (Right Panel)

Module: `vga_status_overlay`
Inputs:

- Background RGB from range plot.
- `samp_q15`, `fan_temp_on`, `fan_manual_on`, `fan_pir_on`.
- `duty_final_q15_dbg`, `pir_active`, UART counters, `theta_q15`, encoder state.

Widgets:

- Temperature bar + numeric readout.
- Fan tiles (TEMP / MANUAL / PIR) lighting when active.
- Fan duty slider.
- PIR motion streak / activity bar.
- UART activity tiles / byte counts.
- Angle / heading indicators.
- Rotary spinner indicator (logical angle).

5.5 Rotary Vertical Bar Overlay

Module: `rotary_bar_overlay`

- Applied last, after HUD composition.
- Draws a vertical side bar whose height $\propto |$`enc_pos`$|$.
- Bar color encodes direction (CW → green-ish, CCW → red-ish).
- Includes an "activity decay" so the bar fades when idle.

6. UART Telemetry: What the PC Sees

6.1 Frame Format (17 Bytes)

Index	Content
0	0x55 (sync 0)
1	0xAA (sync 1)
2–5	`timestamp` [31:0] (little-endian)
6–7	`theta_q15` (angle, Q1.15turns)
8–9	`dist_mm` (range in mm)
10–11	`temp_q15` (Q1.15temperature)
12–13	`duty_q15` (Q1.15fan duty)
14	`status` (compressed flags)
15–16	CRC16 (frame integrity)

6.2 Field Interpretation

- **timestamp**: free-running microsecond-scale counter.
- **theta_q15**: angle in turns; convert to radians or degrees as above.
- **dist_mm**: direct distance in millimetres.
- **temp_q15**: canonical temperature code.
- **duty_q15**: final fan command (0.0 = off, 1.0 = full duty).
- **status**: encodes ToF state (valid/saturated/error) + system flags.

7. MATLAB / PC Tools

7.1 Telemetry Decoder

MATLAB (or Python/C) script:

- Opens serial port at 2 Mbit/s, 8-N-1.

- Searches the stream for 0x55, 0xAA preamble.
- Extracts each 17-byte frame; checks the CRC.
- Writes CSV rows: `idx, timestamp_s, theta_deg, dist_mm, temp_q15, duty_q15, status_hex, crc_ok`.

7.2 Live Plots

Typical plots:

- Distance vs. time.
- Polar scatter: `[x,y] = pol2cart(theta_rad, dist_mm)`.
- Status timeline (error/valid flags vs. frame index).

7.3 Logo/Image Streaming

MATLAB helpers:

- Resize arbitrary image to 320×240.
- Quantize to RGB444 and pack into bytes.
- Stream pixels to FPGA over UART into inactive logo buffer.
- Trigger a buffer swap to make the new image visible.

8. Memory Architecture

8.1 ToF Framebuffer

- Size: 256×256 bits (1 bit per pixel).
- Dual-port BRAM:
  - Port A @ 100 MHz for writer + clear FSM.
  - Port B @ 25 MHz for VGA colorizer.
- Cleared once per sweep via `sweep_wrap`.

8.2 Double-Buffered Logo Memory

- Two banks: 320×240 RGB444 each.
- System domain (100 MHz): UART frame loader writes new image into `write_buf_sys`.
- Pixel domain (25 MHz): `display_buf_pix` selects which bank is drawn.
- `logo_swap_req_sys` coordinates a clean bank toggle on `frame_tick`.

9. Key Signals Cheat Sheet

Signal	Meaning
`dist_mm`	ToF distance (mm)
`theta_q15`	Survey angle (Q1.15turns)
`samp_q15`	Board temperature (Q1.15)
`duty_final_q15_dbg`	Final fan duty command (Q1.15)
`pir_active`	Recent PIR motion (level)
`pir_rise`	New motion event (pulse)
`rot_pos`	Rotary encoder position (signed)
`step_pulse_src`	Unified survey step (AUTO/MANUAL)
`frame_tick_pix`	Top-left of active VGA frame (1 pulse/fr
`logo_swap_req_sys`	Request to swap logo buffers
`logo_display_buf_pix`	Which logo bank is on-screen

10. What to Watch During the Demo

On the VGA screen:

- **Top-left:** logo viewport (static or streamed from PC).
- **Left-half:** live ToF range map (green hits, gray background, cyan rings, white axes).
- **Right panel:** HUD with temperature bar, fan tiles, PIR motion streak, UART indicators, angle widgets.
- **Far right:** rotary bar overlay reacting to manual spin.

On the board LEDs:

- LED9: ToF interrupt activity.
- Selected LEDs: fan contribution flags (TEMP / MANUAL / PIR).
- Stream LEDs: UART byte activity and simple bargraph.
- RGB LED: PIR motion decay (how long since someone moved).

On the PC:

- Live plots of distance vs. time and polar maps.
- CSV logging of all frames for offline analysis.
- Optional logo/image streaming into the VGA logo viewport.