

ECE 215/S – Principles of Electronic Cyber Warfare

Block 2: Signal Processing

Lesson 27: Project 2 Workday 2

Department of Electrical and Computer Engineering
United States Air Force Academy

Fall 2026

Today's Plan

- **Full period** of work time — instructors circulating
- Bring problems to us **early**. A question at the ten-minute mark is worth ten at the end
- **Quiz 7** (Objectives 2.7–2.8) is due tonight
- **Project 2** is due at Lesson 29 — **check Gradescope for the exact deadline**

This is your last scheduled work period. Plan the rest accordingly.

Today's Goal: The Outputs

Last lesson was **transitions** — the state name in the status line moving when it should. Today is **outputs**, and then the demo.

- **Moore:** the telemetry voltage on `DAC1`, verified **on the DMM** at all three states
- **Mealy:** FIRE — `*** MISSILE AWAY ***`, the LED strobe, and **both** refusal messages
- **Then rehearse the demo script**, end to end, at least twice

If your transitions are not solid yet, fix those first. An output driven off a wrong state is not a debugging problem you can see.

The Moore Output: the Due's DAC Does Not Reach the Rails

$$V_{\min} = \tfrac{1}{6}(3.3\,\text{V}) = 0.55\,\text{V} \qquad V_{\max} = \tfrac{5}{6}(3.3\,\text{V}) = 2.75\,\text{V}$$

- `analogWrite(DAC1, 0)` puts out about $\mathbf{0.55}$ V, **not** 0 V. Anything you ask for outside 0.55–2.75 V simply never appears at the pin
- `analogWrite()` defaults to **8 bits** — 256 codes across a 2.20 V span. 2.4 V lands on **code 214** $= 2.396$ V, so expect **2.39–2.40** on the meter — anything inside ± 0.02 V is correct
- That is *why* the three telemetry voltages are 0.8, 1.6 and 2.4 V: all three sit comfortably inside the span the hardware can actually produce
- `dacWriteVolts()` is provided and does the mapping and the clamp. **Your job is to call it with the right voltage for the state** — and nothing else

Put the DMM on it and confirm all three. A number in your code is not a volt on a pin.

The Mealy Output: FIRE and Its Interlocks

FIRE is asserted only when **all four** are true on the same tick:

state $=$ LOCK	the seeker is holding the target
P	the operator pressed the pickle
E	the target is inside the engagement envelope
M	the master arm plug is inserted

- Then: `Serial.println("*** MISSILE AWAY ***")` and one call to `flashLaunchIndicator()`, so the LED strobes **on that tick only**
- And the machine returns itself to **SEARCH** — the target has been serviced

It depends on the state *and* on this tick's inputs. Hold that for Question 1.

Refused Pickles: Say Why, Change Nothing

A pickle in LOCK that fails an interlock is **not a transition**:

Condition (in priority order)	The scope prints
LOCK $\cdot P \cdot D \cdot \overline{E}$	`NO FIRE -- OUTSIDE ENVELOPE`
LOCK $\cdot P \cdot D \cdot \overline{M}$	`NO FIRE -- MASTER ARM SAFE`

- **Both denials carry** D — on the tick the target vanishes, break lock and say **nothing**
- The machine **stays in LOCK**, and `DAC1` stays at 2.4 V
- **Check the envelope first, then the master arm** — so a pickle with both interlocks open reports the envelope
- Prefer the other order? Fine — but then say so, and **be consistent across your diagram, your table, and your code**

Debugging: Isolate the Layer Before You Fix It

- **Read the status line first.** It already prints the state, the distance, and D, E, `ARM`. Most "the FSM is broken" reports are an input that is not what the team assumed
- `d= --- cm` **means** `CAL_K` **is still** `0.0`**.** The machine cannot detect anything and will sit in SEARCH forever
- `D=0 E=1` **is impossible.** $6.0 \leq 12.0$, so the envelope is inside the gate — your two thresholds are swapped
- **A key that "did not work"** probably did: one-shots last exactly one tick, and the tick is 2 seconds long. Watch one full line before you press again
- **The DMM reads** 0.55 **V and never moves** $\rightarrow$ Section 4's Moore write is still empty — **or the probe is on** `DAC0`

Status line, then transitions, then outputs. Never debug two layers at once.

The Demo Script (1 of 2)

Your instructor will ask you to fly this engagement, and **both of you** should be able to narrate it:

1. **Cold start.** Nothing in front of the sensor. `SEARCH`, DMM 0.8 V, plug **out**
2. **Acquire.** Target in to about 10 cm $\rightarrow$ `DETECT`, DMM 1.6 V, `D=1 E=0`
3. **Lock.** Type `'l'` $\rightarrow$ `LOCK`, DMM 2.4 V
4. **Refused pickle — SAFE.** Plug still out; bring the target inside 6 cm so `E=1`, type `'p'` $\rightarrow$ `NO FIRE -- MASTER ARM SAFE`. Still LOCK, still 2.4 V
5. **Arm.** Insert the plug. `ARM=1`. Nothing else changes
6. **Refused pickle — outside the envelope.** Back out to about 10 cm so `E=0`, type `'p'` $\rightarrow$ `NO FIRE -- OUTSIDE ENVELOPE`. Still LOCK
7. **Close and fire.** Back inside 6 cm, type `'p'` $\rightarrow$ `*** MISSILE AWAY ***`, the LED strobes, the machine returns itself to **SEARCH**, and the DMM follows to 0.8 V **on the next tick** (the shot tick still reads 2.4 — Moore)

The Demo Script (2 of 2) — the Loss Edges

8. **Target loss from DETECT.** Let the machine reacquire to DETECT, then wave the target away → **SEARCH** on the next tick
9. **Target loss from LOCK.** Reacquire, type `'l'` to get back to LOCK, then wave the target away → **SEARCH**, break lock

- Both edges are **in the portal demo** too: if your board disagrees with it on the same inputs, **your transitions are wrong**. The **diagram**, the **table** and the **code** are still yours, loss edges included

After the shot the target is still there, so the next tick reacquires to DETECT. That is correct — be ready to explain it, not to "fix" it.

The Four Discussion Questions

Answered **individually**, in the Gradescope assignment **Project 2 Discussion Questions**. Every answer is a multiple-choice pick or a number, and Gradescope grades it automatically:

1. **Moore, Mealy, or both?** For the `DAC1` telemetry voltage and for FIRE, pick the classification *and* its justification, checked against your Section 4: *what does each output function need to be told?*
2. **What can the DAC actually put out?** Its real span is 0.55–2.75 V at 8 bits: the step size ΔV, the DAC steps between adjacent telemetry voltages, and what the DMM on `DAC1` would read if the design had said 0.0 / 1.0 / 3.0 V
3. **What is one ADC count worth at the detection gate?** With the datasheet pair `CAL_K` $= 6.20$ cm·V, `CAL_B` $= -0.75$ cm (not your own), and $\Delta V = 3.22$ mV, how many cm is one count at $d = 12.0$ cm? Then: which limits the detection decision more, **quantization** or **calibration bias** (the lab held $|\text{error}| \leq 0.5$ cm), and by roughly what factor?
4. **Why must the illegal state fail to SEARCH?** Delete the guard and land the machine on `11`: what does the starter's `nextState()` do on the ticks that follow? And why force **SEARCH** rather than DETECT or LOCK?

Submission Checklist

- ☐ Arming plug wired — jumper from pin 2 to `GND` — and the DMM on `DAC1` and `GND`
- ☐ `CAL_K` and `CAL_B` from the Lesson 22 lab plugged in and `distanceCm()` completed
- ☐ **State transition diagram** drawn — three states, **both target-loss edges**, FIRE on its arrow
- ☐ **Transition table** completed — every state against every input condition, in priority order
- ☐ Transitions written in Section 3, with the **illegal-state guard left in place**
- ☐ Moore telemetry verified on the DMM — $0.8\,\text{V}$, $1.6\,\text{V}$, $2.4\,\text{V}$
- ☐ Mealy FIRE verified — `MISSILE AWAY`, the LED strobe, and **both** refusal messages
- ☐ Both target-loss edges demonstrated — one from DETECT and one from LOCK
- ☐ Full demo script run for your instructor, **both partners** walking through code
- ☐ Documentation statement — collaboration, internet resources, and any GenAI use
- ☐ One PDF: the diagram, the documentation statement, **both names**
- ☐ **Group** submission to **Project 2** on Gradescope: `.ino`, `.xlsx`, PDF; autograder report read
- ☐ **Individual** submission to **Project 2 Discussion Questions** on Gradescope

Wrap-Up

What Project 2 covered:

- A **transducer** read through an **ADC**, calibrated and conditioned by you (Objectives 2.5, 2.6)
- An **FSM** with a **Moore** output and a **Mealy** output, an illegal-state guard, and interlocks (Objective 2.8)
- You built the **decision** half of a fire-control system. **Block 3 builds the detection physics** — the radar range equation, SNR, and the jamming that attacks the comparison directly. Before a seeker can compare signal against noise it has to **put the signal on a carrier and get it back** — that is Lessons 28–30

Due tonight: **Quiz 7** (Objectives 2.7–2.8).

- **Lesson 28 — Block 3 begins: modulation and AM. Project 2 is due at Lesson 29** — exact deadline on Gradescope