

# Plan Maestro v1 — TMRCU ADC

Roadmap F1→F4, WPs, KPIs y Artefactos

Proyecto TMRCU / MSL

2 de septiembre de 2025

## Resumen ejecutivo

Este documento empaqueta los entregables de ingeniería del programa TMRCU-ADC: *checklist* F1 para SYNCTRON/ΣFET (SHNO), *netlist* & Σ-IR con plantilla Synk, y el POC de Kuramoto (32×32). Incluye **tablas de paquetes de trabajo (WPs)**, **KPIs/gates**, y **esquemas**; además, **schemas YAML/JSON** quedan embebidos como apéndices para trazabilidad.

## 1. Roadmap y fases

**Diagrama (PDF):** Diagrama\_Roadmap\_ADC\_TMRCU.pdf

*Nota:* Para compilar con la figura, coloque el PDF en el mismo directorio y use:  
`\includegraphics[width=\textwidth]{Diagrama_Roadmap_ADC_TMRCU.pdf}`

## 2. Paquetes de trabajo (WPs)

Fase	WP	Descripción	Artefactos
F1	WP1.1	Diseño SHNO ( $\mu, K$ ) y layout CPW $50\Omega$	Stack, máscaras, DRC
	WP1.2	Fabricación P0 (nanoconstricción HM/FM)	Wafer/die
	WP1.3	Banco RF: DC, VNA/SA, lock-in, Bias-T, SOLT	Fixture, scripts
	WP1.4	Ensayo: $\Sigma(u_g)$ , histéresis, locking	Datos, ajuste SL
F2	WP2.1	Celdas: CΣA/CΣS/CΣD/Latch Σ-SR	Biblioteca
	WP2.2	Bus Σ y re-phase; P&R con pérdidas	P&R rules
	WP2.3	Matriz $32 \times 32$ (bloques $8 \times 8$ )	Die P2
	WP2.4	POC Kuramoto/Ising	Bench, scripts
F3	WP3.1	Lenguaje Synk (tipos, operadores, contratos)	Especificación
	WP3.2	Σ-IR (K,timing,placement,seguridad)	Schema JSON
	WP3.3	Σ-OS (re-phase, telemetría, failsafe)	Runtime
	WP3.4	Toolchain (compilador, simulador, profiler)	Tooling
F4	WP4.1	Dispositivo edge SAC + biosensores	BOM, CAD
	WP4.2	CSL-H en Synk (multiescala)	Modelos
	WP4.3	Ensayos preregistrados	Protocolo
	WP4.4	Validación clínica/ética	Dossier

## 3. KPIs y gates (falsables)

KPI	Umbral	Gate	Método
Ajuste $\Sigma(u_g)$ a Stuart–Landau	RMSE < 0.1	F1→F2	Barrido $u_g$ ; IC95 % parámetros
Injection locking estable	rango captura medible	F1→F2	Barrer $\omega_{in} \pm 200$ MHz
Repetibilidad wafer	variación < 10 %	F1→F2	$N \geq 5$ celdas
Celdas (CΣA/CΣS/CΣD)	error < 0.1	F2→F3	$N=500$ corridas
Matriz $32 \times 32$	$\geq 80\%$ nodos en falso; $t < 100\mu s$	F2→F3	Medición $R(t)$
Ventaja MVC	MVC > 100 & IC95 % > 1	F2→F3	Potencia/tiempo GPU vs $\Sigma$
Overhead compilación	< 10 %	F3→F4	Synk→Σ-IR→runtime
Robustez operacional	$\mathbb{P}(x \in \mathcal{C}) \geq 0.99$ (24 h)	F3→F4	Stress, drift térmico
KPIs clínicos (SAC-EMERG)	AUC > 0.85; $\kappa > 0.6$ ; $T_{notify} < 30$ s	F4	Pilotos preregistrados

## 4. Protocolos clave

### F1 — Checklist de laboratorio

Instrumentación (DC, VNA/SA, lock-in, Bias-T, atenuadores, SOLT, de-embedding), scripts (control\_gate, injectar\_coherencia, leer\_salida, analisis\_coherencia), rutina  $\Sigma(u_g)$  con histéresis y locking. Datos crudos + manifest.

### F2 — Biblioteca & $32 \times 32$

$C\Sigma A$  ( $\approx \Sigma_1 \Sigma_2$ ),  $C\Sigma S$  (máx),  $C\Sigma D$  ( $|\Sigma_1 - \Sigma_2|$ ), Latch; bus  $\Sigma$ , re-phase y P&R; POC Kuramoto/Ising.

### F3 — Toolchain

Lenguaje Synk,  $\Sigma$ -IR (,K,placement,timing,seguridad),  $\Sigma$ -OS (planificador, telemetría, failsafe).

### F4 — SAC/CSL-H

Edge con biosensores; ejecución CSL-H; ensayos y KPIs clínicos.

## Apéndice A — Schema $\Sigma$ -IR (JSON)

```
{  
  "target_device": "TMRCU_Processor_v1",  
  "cells": [  
    {  
      "id": "XOR1",  
      "type": "C\u003a3D"  
    }  
  ]  
}
```

## Apéndice B — Manifest de corrida (YAML)

```
run_id: F1_SHNO_YYYYMMDD_NNN  
wafer: W##  
die: D##  
temp_C: 25
```

## Apéndice C — Synk: adder.synk

```
// adder.synk  
function SigmaAdder(A: Sigma, B: Sigma) -> (S: Sigma, C: Sigma) {  
  S = A + B; C = A - B; return (S,C);  
}
```

## Apéndice D — Synk: kuramoto32.synk

```
// kuramoto32.synk
const N = 1024; // 32x32
```