TASKS

1) Hardware and Initial setup

Main Task (Epic)	Subtasks (Action Items)	Dependencies	Notes	Man day
HS-1: Project & Peripherals Setup	**HS-1.1:** Set up STM32CubeIDE project.	None	Use STM32F207ZG target.	0.5
	HS-1.2: Configure clock tree (HSE, PLL).			0.5
	HS-1.3: Configure I2C/SPI for sensor communication (HAL).			0.5
	HS-1.4: Configure GPIO pins for LEDs (HAL).			0.5
	HS-1.5: Configure Ethernet MAC			1

	and related GPIOs (HAL).			
	HS-1.6: Initialize Timer (TIM) for system tick/RTOS (HAL).			1.5

2) Core Drivers and RTOS Integration

Main Task (Epic)	Subtasks (Action Items)	Dependencies	Notes	Man day
DRV-1: Integrate FreeRTOS	**DRV-1.1:** Integrate FreeRTOS into the project.	HS-1.6		1
	DRV-1.2: Create initialization task and start the scheduler.			1
DRV-2: PAJ7660	**DRV-2.1:** Implement sensor power-up and basic I2C/SPI	HS-1.3		1.5

Sensor Driver	initialization sequence.			
	DRV-2.2: Configure sensor to operate in the required Gesture Mode 5.			1
	DRV-2.3: Implement `read_gesture_id()` function (Layer 3 output).	DRV-2.2	Reads the classified integer.	1.5
DRV-3: LED Driver	**DRV-3.1:** Implement `set_led(pin, state)` utility function.	HS-1.4		0.5
	DRV-3.2: Implement `set_leds_by_count(int count)` using the utility.	DRV-3.1	Controls the 5-LED output.	0.5
DRV-4: Networking Stack (LWIP)	**DRV-4.1:** Integrate LWIP library with the Ethernet MAC HAL.	HS-1.5		1.5

	DRV-4.2: Configure LWIP for DHCP address acquisition.			1.5
	DRV-4.3: Implement a basic PING test to verify network connectivity.	DRV-4.2	Debugging check.	0.5

3) Application Logic and Feature Implementation

Main Task (Epic)	**Subtasks (Action Items)**	**Dependencies**	**Notes**	**Man day**
APP-1: Sensing Task	**APP-1.1:** Create RTOS task for continuous sensor polling	DRV-1, DRV-2.3		1
	APP-1.2: Map the sensor's raw IDs to the 5 standard system gestures	APP-1.1		1

	(e.g., ID 8 → Swipe Left).			
APP-2: Local Feedback	**APP-2.1:** Receive Gesture ID from APP-1.2.	APP-1.2		0.5
	APP-2.2: Call **DRV-3.2** to immediately display N LEDs for Gesture ID N.	DRV-3.2	Low-latency path.	0.5
APP-3: MQTT Client & Payload	**APP-3.1:** Integrate lightweight MQTT client library	DRV-4.3		2
	APP-3.2: Implement MQTT connection and authentication logic.	APP-3.1	Handles reconnects.	1.5
	APP-3.3: Implement payload struct: Gesture_ID, Timestamp, Status.	APP-3.1		1

	APP-3.4: Implement **QoS 1** publish logic (requires acknowledgement).	APP-3.3	PRD requirement for reliability.	2.5
APP-4: Failsafe & Monitoring	**APP-4.1:** Initialize and periodically feed the Watchdog Timer.	DRV-1	Ensures system restart on crash.	0.5
	APP-4.2: Monitor sensor read status and network connection status.	DRV-2.3, DRV-4.2		0.5
	APP-4.3: Implement the Failsafe Visual Alert (blink all 5 LEDs) if status check fails.	DRV-3.2, APP-4.2		0.5

4) Quality Assurance and Documentation

Main Task (Epic)	Subtasks (Action Items)	Dependencies	Notes	Man day

QA-1: Performance Testing	**QA-1.1:** Set up timing hooks around sensor read and MQTT publish acknowledgment.	APP-2, APP-3.4		1.5
	QA-1.2: Execute 100 test gestures and calculate average latency.	QA-1.1	Verify $\le 200 \text{ms}$ cap.	1
QA-2: Reliability Testing	**QA-2.1:** Stress test MQTT QoS 1 delivery during simulated network disconnects.	APP-3.4, APP-4.3		1.5
	QA-2.2: Verify Emergency Stop (Closed Fist) response time.	APP-2.2	Needs defined priority.	0.5
DOC-1: Finalizing Documentation	**DOC-1.1:** Document the final MQTT topic structure and JSON/binary payload.	APP-3.3		0.5

	DOC-1.2: Update the architectural diagram and component descriptions to reflect the PAJ7660 integration.	Final QA		0.5