

U3A Computing Group

DATED: 4 September 2026

Version: 1, Outline for discussion

Project Plan: Non-Intrusive Home Activity Monitoring System (NB All references to particular hardware items are purely for example)

1. Project Overview & Objective

This project aims to design, prototype, and test an Internet of Things (IoT) based activity monitoring system for an elderly relative living independently. Using learning from the first project other uses to be determined.

Core Goal

Detect movement and ambient activity patterns within a home without using intrusive cameras, wearable devices, or listening microphones, and send automated notifications or alerts to family members/caregivers in the event of an anomaly or potential emergency.

Target Audience for Development

A small collaborative team of retired IT/engineering specialists exploring hardware, microcontrollers, embedded coding, serverless backends, and simple communication protocols.

2. Key System Requirements & Constraints

- **Privacy & Non-Intrusiveness:** No optical cameras, smart speakers, or mandatory wearables.
- **Ease of Installation:** Battery-operable or simple plug-in devices requiring minimal retrofitting.
- **Reliability & Resilience:** Low power consumption, cellular or local Wi-Fi connectivity, and heartbeat monitoring to detect system offline events.
- **Affordability:** Built using off-the-shelf, low-cost microcontrollers and sensors.

3. Architecture & Hardware Options to Explore

A. Sensing Options (Hardware Discovery Phase)

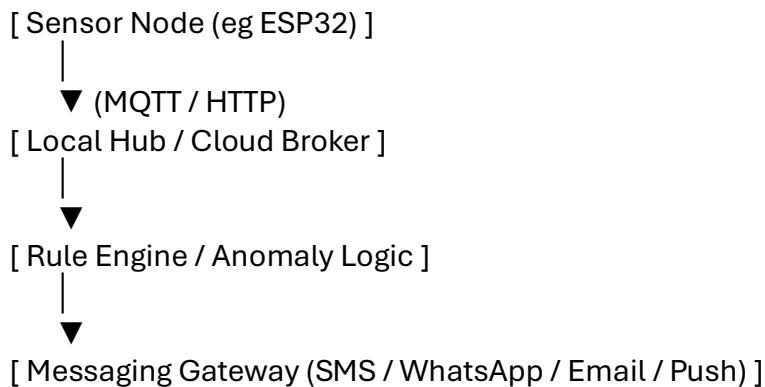
Sensor Type	Potential Use Case	Pros	Cons
PIR (Passive Infrared)	Detect motion across doorways or rooms	Very cheap, highly effective	Only detects motion when moving across field of view
Radar / mmWave (24GHz/60GHz)	Detect presence, micro-movement, breathing	Highly accurate, detects static presence	Requires higher power and tuning
Door/Window Reed Switches	Track front door, fridge, or medicine cabinet openings	Highly reliable, long battery life	Only measures specific contact points

Sensor Type	Potential Use Case	Pros	Cons
Ambient Light / Sound Sensors	Detect light turned on/off or kitchen appliances running	Simple integration	Sensitive to ambient noise/sunlight
Smart Plug / Current Sensor	Monitor energy usage on kettle, TV, or microwave	Clear proxy for daily routine	Requires direct plug access

B. Microcontrollers & Connectivity Options

- **ESP32 / ESP8266:** Built-in Wi-Fi and Bluetooth, low cost, large community support, low-power sleep modes.
- **Raspberry Pi Pico W:** Excellent C/C++ and MicroPython support, low power draw.
- **Cellular / LPWAN (LoRaWAN, NB-IoT):** Backup connectivity if home Wi-Fi is unreliable or unavailable.

4. Software & Cloud Infrastructure Strategy



Key Software Exploration Components

- Firmware Development:**
 - C/C++ via Arduino IDE / ESP-IDF, or MicroPython.
 - Power-management logic (Deep Sleep, interrupt-driven motion wakes).
- Data Ingestion & Messaging Protocol:**
 - **MQTT** (Mosquitto) or **HTTP REST** API.
 - Cloud Platforms: AWS IoT Core, Adafruit IO, Node-RED, or Home Assistant.
- Alerting System:**
 - **SMS/Calls:** Twilio, Telnyx.
 - **Instant Messaging:** Telegram Bot API, WhatsApp Business API, Pushover.
 - **Email:** SendGrid or AWS SES.

4. Workstreams & Phase Breakdown

Phase 1: Hardware & Sensor Discovery (Weeks 1–4)

- Establish baseline environment setups (Arduino IDE / VS Code + PlatformIO).
- Breadboard testing of PIR sensors vs. mmWave radar vs. door contact sensors.
- Evaluate false positives, trigger ranges, and power consumption under deep sleep routines.

Phase 2: System Architecture & Connectivity (Weeks 5–8)

- Configure MQTT broker / cloud endpoint.
- Program microcontrollers to publish motion telemetry and "Heartbeat" messages.
- Develop power optimization routines (battery lifecycle estimation).

Phase 3: Rule Engine & Alert Logic (Weeks 9–12)

- Define routine detection patterns (e.g., "No movement between 07:00 and 10:00 AM" or "Front door open and no return motion within 30 mins").
- Code alerting scripts for normal updates vs. emergency triggers.
- Implement a "Keep-Alive" check to alert caregivers if a node goes offline or loses Wi-Fi.

Phase 4: Prototyping & Enclosures (Weeks 13–16)

- Design or select physical enclosures for sensor nodes.
- Field test in a controlled home environment over a 2-week period.
- Refine thresholds, reduce noise/false alarms, and gather feedback.

5. Roles & Discussion Topics for Team Members

- **Hardware & Circuit Lead:** Sensor selection, pin configuration, power management, custom PCB/perfboard setup.
- **Embedded Firmware Lead:** ESP32/Pico programming, WiFi connection handling, deep sleep logic.
- **Backend & Logic Lead:** MQTT broker management, Home Assistant / Node-RED flows, state management.
- **Communications & Alerting Lead:** Integration with Twilio / Telegram / Push API, fail-safe notification routing.