{\rtf1\ansi\deff0{\fonttbl{\f0 Calibri;}{\f1 Consolas;}}

{\colortbl;\red34\green34\blue34;\red68\green68\blue68;\red0\green120\blue215;}

\fs28\b Autonomous Greeter Robot — Complete Project Document (v1.0)\b0\fs22\par

\par

\b Author:\b0 You + DALL·E (the GPT) \tab \b Date:\b0 2025-10-14\par

\b Purpose:\b0 Design and implementation plan for an autonomous robot that patrols a defined space, detects and recognizes people, greets them, and provides a companion app for live interaction and enrollment.\par

\par\b Table of Contents\b0\par

1. Requirements\par

2. System Specification\par

3. Architecture \& Wiring\par

4. Step-by-Step Build Plan\par

5. Implementation Details\par

6. Testing \& Acceptance\par

7. Schedule (Gantt/PERT)\par

8. Bill of Materials\par

9. Safety Checklist\par

10. Concept Render (SVG)\par

\par\fs24\b 1) Requirements\b0\fs22\par

\b 1.1 Functional\b0\par

1. Autonomous patrol \& navigation within a defined space; obstacle avoidance; physical bumper collision detection with immediate recovery.\par

2. Person detection \& greeting: stop at ~1.0 m; recognized → greet by name; unknown → alternate flow to capture name/purpose.\par

3. Companion app (phone + desktop): live video, two-way audio, remote tele-op, identity guess/override, battery/temps/faults/logs.\par

4. Enrollment \& data management: add/update/remove identities; store embeddings with metadata; consent capture \& audit log.\par

5. Connectivity: BLE for pairing/control/telemetry; Wi-Fi for video/audio and bulk data.\par

6. Safety/fallbacks: physical E-stop \& software stop; low-battery return-to-dock or safe stop; brownout handling.\par

\b 1.2 Non-Functional\b0\par

Privacy-first (consent, local processing where possible, TLS, encryption at rest), reliability (8 h duty), maintainability (modular, clean wiring), cost-conscious (4★+ parts), performance (person detect \<= 500 ms @ 720p, recognition < 1 s, nav loop \>= 20 Hz), and electrical safety compliance.\par

\par\fs24\b 2) System Specification\b0\fs22\par

\b 2.1 Software \& Development Environment\b0\par

• IDE: Visual Studio Code; PlatformIO for MCU; Jupyter for CV experiments.\par

• SBC services (Python): OpenCV, ONNX Runtime, FastAPI, aiortc/GStreamer, SQLite.\par

• MCU (ESP32): C++ via ESP-IDF (preferred) or Arduino framework.\par

• App: Flutter (single codebase for iOS/Android/macOS/Windows/Linux).\par

• Core libs: YOLOv5n/YOLOv8n-face or MediaPipe (detection); ArcFace/MobileFaceNet embeddings (recognition); Vosk or Whisper-small (ASR); Piper/eSpeak NG (TTS). Optional ROS 2 later.\par

\b 2.2 Hardware (value \& reliability)\b0\par

SBC: ODROID-C4 (or RPi 4/5).\par

MCU: ESP32-WROOM dev board.\par

Chassis: 4WD kit (metal/acrylic) with motors (+encoders recommended).\par

Motor drivers: 2× TB6612FNG (TT motors) \i or\i0 2× Cytron MD10C for higher current.\par

Sensors: 3–4× VL53L1X ToF, 2× Ultrasonic, 3–4× bumper switches, IMU (BNO055 or MPU-6050).\par

Camera: 1080p USB webcam (90–120° FOV). Audio: USB mic + 3–5 W speaker + PAM8403 amp.\par

Battery: 3S LiPo 11.1 V, 2200–5000 mAh (XT60). Power: 5 V 6 A buck; 3.3 V LDO. Safety: 20 A fuse, master switch, LiPo charger.\par

\b 2.3 Power System\b0\par

3S LiPo → XT60 → 20 A fuse → master switch → distribution.\par

Rails: 12 V (motors/drivers), 5 V/6 A buck (SBC/USB), 3.3 V LDO (sensors/ESP32). Star ground; twist motor leads; decoupling near drivers; 0.1 µF across motors.\par

\par\fs24\b 3) Architecture \& Wiring\b0\fs22\par

\b High-level data/control\b0\par

{\f1

[USB Cam] --> [SBC: CV/ASR/TTS] <--USB--> [Mic/DAC/Amp/Speaker]\par

| | \par

| | \-- Wi-Fi/WebRTC --> [Companion App]\par

| \-- BLE/Wi-Fi --> [ESP32 MCU]\par

|\par

[SQLite: embeddings, logs]\par

\par

[ESP32] <--> [Motor Drivers] --> [4x Motors]\par

| \-- [Encoders]\par

| \-- [ToF + Ultrasonic]\par

| \-- [Bumpers]\par

\-- [IMU (I2C)]

}{\f0\par}

\b Power (ASCII)\b0\par

{\f1

[3S LiPo]--XT60--(20A Fuse)--[Master Switch]---+\par

+---[5V/6A Buck]----> SBC + USB\par

+---[12V Rail]------> Motor Drivers/Motors\par

+---[3.3V LDO]------> ESP32 + Sensors\par

All Grounds -> Star Point -> Chassis Ground

}{\f0\par}

\b Example Pin Map (ESP32)\b0\par

TB6612 #1 (Left): AIN1 GPIO18, AIN2 GPIO19, PWMA GPIO5; BIN1 GPIO16, BIN2 GPIO17, PWMB GPIO4.\par

TB6612 #2 (Right): AIN1 GPIO26, AIN2 GPIO27, PWMA GPIO25; BIN1 GPIO14, BIN2 GPIO12, PWMB GPIO13.\par

Encoders: LF A/B GPIO32/33; LR A/B GPIO34/35; RF A/B GPIO36/39; RR A/B GPIO21/22 (adjust to interrupt-capable pins).\par

I2C: SDA GPIO23, SCL GPIO22. Ultrasonic: Trig GPIO2, Echo GPIO15 (use divider if 5 V echo).\par

\par\fs24\b 4) Step-by-Step Build Plan\b0\fs22\par

1) Mechanical assembly.\par

2) Power \& safety bring-up (fuse, switch, rails, ripple check).\par

3) Motor subsystem wiring \& bench test (PWM sweep each wheel).\par

4) Sensors wiring \& validation (IMU, ToF, ultrasonic, bumpers).\par

5) SBC setup (OS, Python, OpenCV, ONNX Runtime, FastAPI, aiortc/GStreamer, SQLite).\par

6) Comms (BLE GATT on ESP32; Wi-Fi/WebRTC on SBC).\par

7) Control loop integration (ESP32 PID; SBC high-level planner @ 20–50 Hz).\par

8) Perception (detector + tracker + face embeddings + gallery).\par

9) Dialogue (VAD, ASR, intent FSM, TTS; operator override).\par

10) App (Flutter: Connect, Live View, Controls, People, Logs).\par

11) Integration tests (latency, FPS, thresholds, battery, failovers).\par

12) Field tuning (PID, stop distance, ASR params, audio levels).\par

\par\fs24\b 5) Implementation Details\b0\fs22\par

\b 5.1 Locomotion (ESP32, C++)\b0\par

4WD skid-steer; PID per wheel @ 50–100 Hz; watchdog (stop if no cmd in 500 ms); bumper ISR (reverse+turn).\par

{\f1

void loop(){\par

static uint32\_t last=millis();\par

if(millis()-last>=20){ last=millis();\par

readEncoders(); estimateWheelSpeeds();\par

if(bumperHit()) planEvasive();\par

applyPID();\par

if(staleCommand(500)) stopAll();\par

publishTelemetry();\par

}\par

}

}{\f0\par}

\b 5.2 High-Level Planner (SBC, Python)\b0\par

Reactive rules: ToF < 0.40 m → slow; < 0.25 m → stop+rotate away. If person track present, stop at 1.0 m, then run face pipeline.\par

\b 5.3 Person Detection \& Recognition (SBC, Python)\b0\par

Flow: frame → detect faces → track → align → embed → compare (cosine).\par

Thresholds: < 0.45 match; 0.45–0.60 uncertain → ask operator; > 0.60 unknown.\par

{\f1

det = OnnxDetector("yolov8n\_face.onnx")\par

emb = OnnxEmbedder("arcface\_r50.onnx")\par

db = EmbeddingIndex("faces.sqlite")\par

}{\f0\par}

\b 5.4 Speech (ASR/TTS) \& Dialog\b0\par

ASR: Vosk (offline) or Whisper-small if SBC allows. VAD: WebRTC. TTS: Piper. Dialog: small FSM (Greeting → Known/Unknown → HelpOptions → Goodbye). No raw audio stored unless consented.\par

\b 5.5 Networking\b0\par

BLE GATT (cmd/telemetry/sensors). WebRTC over Wi-Fi (H.264/VP8+Opus, DTLS-SRTP). REST/WS for enrollment, logs, config. Security: LAN-only by default, TLS, pre-shared key or local OAuth.\par

\b 5.6 Data \& Consent (SQLite)\b0\par

Tables: people (id, name, consent\_ts), embeddings (vector blob), logs (ts,event,meta). Enrollment requires explicit consent checkbox.\par

\par\fs24\b 6) Testing \& Acceptance\b0\fs22\par

Unit: PID stability, sensor drivers, ROC for recognition.\par

Integration: obstacle course, 50 approach trials (stop distance stats).\par

Performance: detector FPS \>= 15; recognition \<= 1 s; greet \<= 1.5 s.\par

Battery: runtime to 20% SOC; thermal check under sustained load.\par

Safety: E-stop responsiveness; fuse trip; brownout resilience.\par

\par\fs24\b 7) Schedule \& Deliverables\b0\fs22\par

\b 7.1 Mermaid Gantt (for any Mermaid viewer)\b0\par

{\f1

gantt\par

dateFormat YYYY-MM-DD\par

title Autonomous Greeter Robot - Plan\par

section Hardware\par

Chassis & Power :done, H1, 2025-10-15, 7d\par

Motors & Drivers :active, H2, 2025-10-22, 5d\par

Sensors & Encoders : H3, 2025-10-27, 4d\par

section Firmware (ESP32)\par

Motor Control + PID : F1, 2025-10-31, 6d\par

Sensors + Telemetry : F2, after F1, 5d\par

BLE GATT : F3, after F2, 4d\par

section SBC & Perception\par

OS & Tooling : S1, 2025-11-15, 3d\par

Video/Audio Pipeline : S2, after S1, 5d\par

Person Detect + Track : S3, after S2, 6d\par

Face Recognition + Gallery : S4, after S3, 6d\par

Speech (ASR/TTS) : S5, after S2, 4d\par

section App\par

Flutter scaffold + BLE : A1, 2025-11-05, 6d\par

WebRTC live view : A2, after A1, 5d\par

Enrollment & Consent : A3, after A2, 5d\par

section Integration\par

High-level Planner : I1, after F3 S4, 5d\par

End-to-end Greeter Demo :milestone, M1, after I1, 2025-12-05, 0d\par

Field Testing & Tuning : I2, after M1, 7d

}{\f0\par}

\b 7.2 PERT-friendly CSV\b0\par

{\f1

ID,Task,Duration(d),DependsOn\par

H1,Chassis & Power,7,\par

H2,Motors & Drivers,5,H1\par

H3,Sensors & Encoders,4,H1\par

F1,Motor Control + PID,6,H2\par

F2,Sensors + Telemetry,5,F1\par

F3,BLE GATT,4,F2\par

S1,OS & Tooling,3,H1\par

S2,Video/Audio Pipeline,5,S1\par

S3,Person Detect + Track,6,S2\par

S4,Face Recognition + Gallery,6,S3\par

S5,Speech (ASR/TTS),4,S2\par

A1,Flutter scaffold + BLE,6,S1\par

A2,WebRTC live view,5,A1\par

A3,Enrollment & Consent,5,A2\par

I1,High-level Planner,5,F3;S4\par

M1,End-to-end Greeter Demo,0,I1\par

I2,Field Testing & Tuning,7,M1

}{\f0\par}

\par\fs24\b 8) Bill of Materials (BOM)\b0\fs22\par

ODROID-C4 + 32–64 GB microSD; ESP32 dev board; 4WD chassis (with motors+encoders); 2× TB6612FNG (or 2× Cytron MD10C); 3–4× VL53L1X ToF, 2× Ultrasonic, 3–4× bumper switches, 1× IMU; 1080p USB webcam; USB mic; small speaker + PAM8403 amp; 3S LiPo 2200–5000 mAh; XT60; 20 A fuse; master switch; 5 V/6 A buck; 3.3 V LDO; wiring kit; ferrules; screw terminals; standoffs; zip ties; LiPo charger \& safe bag.\par

\par\fs24\b 9) Safety Checklist\b0\fs22\par

Correct fusing; wire gauges (motor 16–18 AWG, logic 22–24 AWG); strain relief; insulated XT60; EMI caps (0.1 µF) on motors; twisted motor pairs; star ground; charge LiPos off-robot; never unattended; store in LiPo bag.\par

\par\fs24\b 10) Concept Render (SVG)\b0\fs22\par

The concept image is provided as SVG text in the project files. Save the SVG to \i robot-concept.svg\i0 and open in any browser or vector editor.\par

\par\pard\sa240\qc\fs18\cf2 --- End of Document ---\cf0\fs22\par

}