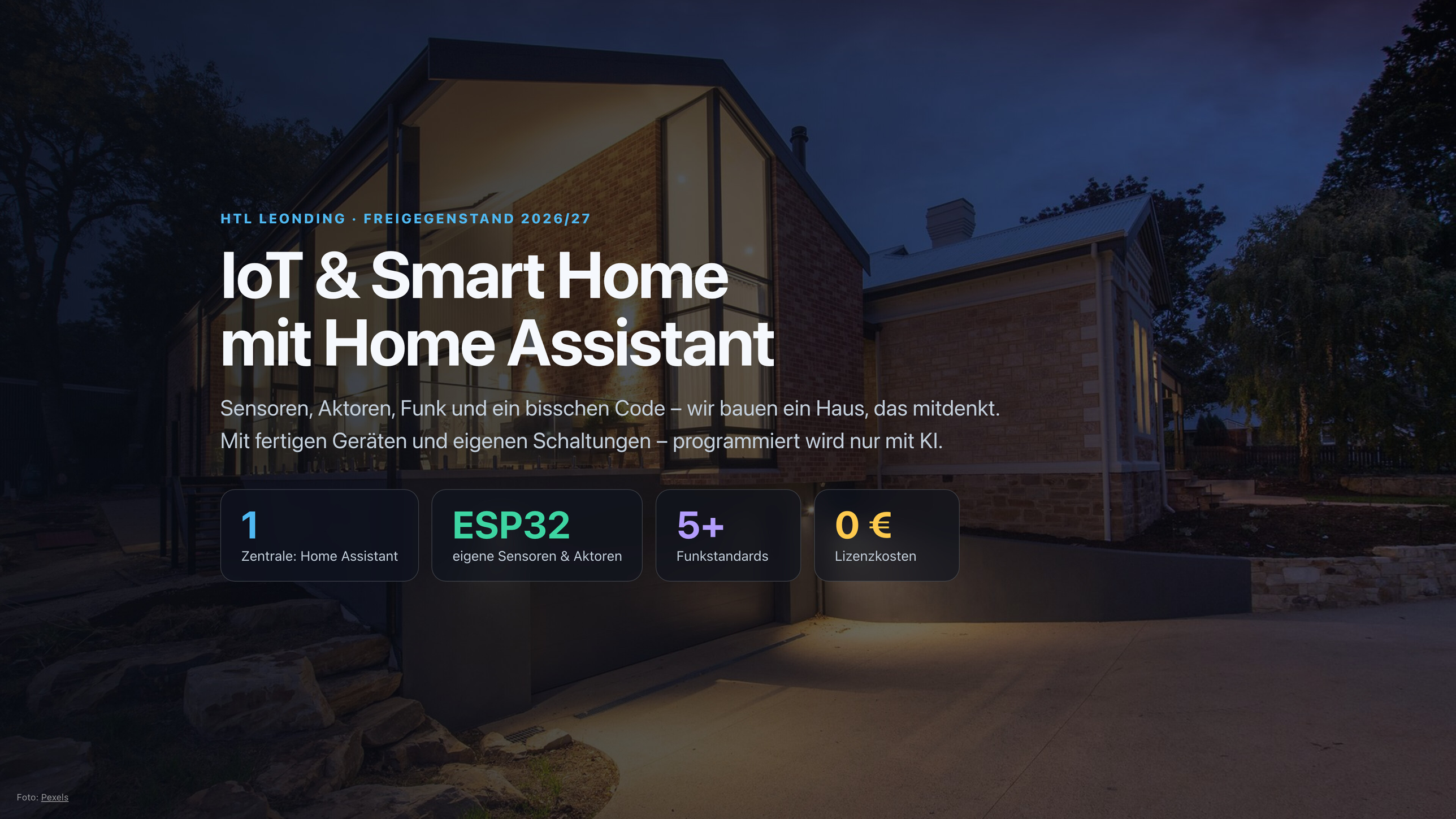


HTL LEONDING · FREIGEGENSTAND 2026/27

IoT & Smart Home mit Home Assistant

Sensoren, Aktoren, Funk und ein bisschen Code – wir bauen ein Haus, das mitdenkt.
Mit fertigen Geräten und eigenen Schaltungen – programmiert wird nur mit KI.

1	ESP32	5+	0 €
Zentrale: Home Assistant	eigene Sensoren & Aktoren	Funkstandards	Lizenzkosten

Foto: Pexels

DIE ZENTRALE

Home Assistant – alles an einem Ort

Open Source, läuft lokal, herstellerneutral. Über 3.000 Integrationen verbinden Geräte, die sonst nie miteinander reden würden.

🔌 Integrationen

IKEA, Shelly, Hue, Aqara, Wetter, Strompreis …

⚙️ Automationen

Auslöser → Bedingung → Aktion, grafisch oder YAML

📊 Dashboards

Am Handy, Tablet an der Wand, im Browser



Home Assistant

Raspberry Pi · HA Green · Mini-PC

🧩 Add-ons

ESPHome, Zigbee2MQTT, Mosquitto (MQTT), Matter-Server

⚡ Energie

Verbrauch, PV-Ertrag, Kosten – pro Gerät

🗣️ Assist

Sprachsteuerung – auch komplett offline

Grundregel: Das Haus muss auch ohne Internet und ohne Hersteller-Cloud funktionieren.

Foto: Pexels

SO ARBEITEN WIR

Ihr tippt keinen Code – die KI programmiert

Programmierung und Konfiguration erledigen wir **ausschließlich mit KI**. Ihr beschreibt, was passieren soll – die KI schreibt ESPHome-Code, Automationen und Dashboards.

💬 **Ihr beschreibt**
in ganz normalen Sätzen

→

🤖 **KI erstellt**
Code, Konfiguration, Verdrahtungsplan

→

🔍 **Ihr prüft**
testen, verbessern, nachfragen

→

🏠 **Es läuft**
in Home Assistant

💬 Zum Beispiel

> *„Bau mir eine CO_2-Ampel mit ESP32 und SCD40: grün unter 800 ppm, gelb bis 1.200, rot darüber. Schick mir eine Nachricht, wenn es rot wird.“*

→ Die KI liefert Verdrahtung, ESPHome-Datei und Automation.

🎓 Was ihr trotzdem lernt

- **Verstehen** statt abtippen – was macht der Code?
- **Prüfen** – auch die KI irrt sich
- **Fehler finden** – Logs lesen, gezielt nachfragen
- **Gut beschreiben** – je genauer, desto besser das Ergebnis

Foto: Pexels

KAUFEN STATT LÖTEN

Fertige Sensoren & Aktoren

Günstig, zertifiziert, sofort einsatzbereit – Home Assistant bindet sie lokal ein.

🛋️ IKEA

Zigbee · Matter · Thread

- **Lampen** – TRÅDFRI / KAJPLATS
- **Luftqualität** – VINDSTYRKA, ALPSTUGA
- **Tür/Fenster**, **Bewegung**, **Wasserleck**
- **Steckdose** mit Strommessung

🔷 Shelly

WLAN · BLE · Matter

- **Relais** für die Unterputzdose (1PM, 2PM)
- **Plug S** – Zwischenstecker mit Messung
- **Pro 3EM** – Energiezähler für das ganze Haus
- **H&T**, **BLU Door/Window**, **Dimmer**

💡 Philips Hue · Aqara · Sonoff

Zigbee · Thread · WLAN

- **Hue** – Lampen, Bewegungsmelder
- **Aqara** – Mini-Sensoren, FP2 Präsenz-Radar
- **Sonoff** – Zigbee-USB-Stick, Schalter
- Thermostate, Rauchmelder, Kameras …

📥 Sensoren messen

🌡️ Temperatur · 💧 Feuchte · 🚪 Kontakt · 🏃 Bewegung · 🫁 CO_2 / Feinstaub · ⚡ Strom · 🌊 Wasserleck

📤 Aktoren handeln

💡 Licht · 🔌 Steckdose · 🪟 Rollladen · ♨️ Thermostat · 🔔 Sirene · 🔒 Schloss

Foto: Pexels

SELBST GEBAUT

Eigene Sensoren & Aktoren mit ESP32

Ein Mikrocontroller um ein paar Euro, WLAN und Bluetooth inklusive. Konfiguriert wird mit **ESPHome** – die YAML-Datei dazu schreibt die KI.

📥 Sensoren

- **BME280** – Temperatur, Feuchte, Druck
- **SCD40** – echtes CO_2
- **LD2410** – mmWave-Radar, erkennt Anwesenheit
- **Bodenfeuchte** – kapazitiv
- **VL53L0X** – Abstand per Laser
- **DS18B20** – wasserdichtes Thermometer

📤 Aktoren

- **Relais** – schaltet Pumpe, Lüfter
- **WS2812B** – adressierbare LED-Streifen
- **Servo** & Schrittmotor
- **Buzzer** – akustischer Alarm
- **OLED / E-Paper** – kleine Anzeigen
- **IR-Sender** – steuert alte Geräte

```yaml
# co2-ampel.yaml
esphome:
  name: co2-ampel
esp32:
  board: esp32dev
wifi: !secret wifi
api:        # direkt zu Home Assistant
ota:        # Updates per WLAN

i2c:
sensor:
  - platform: scd4x
    co2:
      name: "CO2 Klasse"
```

Von der KI geschrieben. Flashen, einstecken – der Sensor taucht automatisch in Home Assistant auf.

Foto: Pexels

KOMMUNIKATION

Wie reden die Dinge miteinander?

ANWENDUNG	**Matter** gemeinsame Sprache aller Hersteller – lokal, ohne Cloud			**MQTT · ESPHome-API** Nachrichten an Home Assistant	
NETZWERK / FUNK	📶 **WLAN**	🧵 **Thread**	🐝 **Zigbee**	🔵 **Bluetooth LE**	📻 **433 MHz**

STANDARD	FREQUENZ	TOPOLOGIE	STROMVERBRAUCH	TYPISCH FÜR	BRAUCHT
📶 **WLAN**	2,4 / 5 GHz	Stern (Router)	hoch	Shelly, ESP32, Kameras	Router
🐝 **Zigbee**	2,4 GHz	Mesh	sehr niedrig	IKEA, Hue, Aqara – Batteriesensoren	Koordinator (USB-Stick)
🧵 **Thread**	2,4 GHz	Mesh, IPv6	sehr niedrig	neue Matter-Geräte	Border Router
🔷 **Matter**	läuft über WLAN, Thread oder Ethernet		–	herstellerübergreifend	Matter-Controller (HA)
🔵 **Bluetooth LE**	2,4 GHz	Punkt-zu-Punkt	niedrig	Thermometer, Anwesenheit	ESP32 als BT-Proxy
📻 **433 MHz**	433 MHz	nur senden	minimal	Wetterstationen, Funksteckdosen	SDR-Stick – unverschlüsselt

Merke: **Matter ist kein Funk**, sondern eine gemeinsame Sprache. **Thread** ist der stromsparende Funk darunter.

Foto: Pexels

AUTOMATIONEN

Wenn … und … dann – das Haus handelt selbst

⚡
Auslöser
CO_2 steigt über 1.000 ppm

→

🤔
Bedingung
Fenster ist zu, Unterricht läuft

→

🎬
Aktion
Ampel rot + Nachricht „Bitte lüften!“

⚡ Mögliche Auslöser

- 🕐 Uhrzeit
- 🌅 Sonnenauf-/-untergang
- 📈 Messwert über/unter
- 🚪 Zustand ändert sich
- 📍 Jemand kommt heim
- 🔘 Tastendruck
- 🌐 Webhook

💡 Ideen

- **Gang**: Bewegung + dunkel → Licht 2 Minuten an
- **Heizung**: Fenster länger als 5 min offen → Thermostat aus
- **Sonne**: PV-Überschuss → Steckdose für Ladegerät ein
- **Pflanze**: Erde zu trocken → Pumpe 10 Sekunden an

```yaml
# automations.yaml
- alias: Lüften-Erinnerung
  triggers:
    - trigger: numeric_state
      entity_id: sensor.co2_klasse
      above: 1000
      for: "00:05:00"
  conditions:
    - condition: state
      entity_id: binary_sensor.fenster
      state: "off"
  actions:
    - action: light.turn_on
      target: { entity_id: light.ampel }
      data: { color_name: red }
    - action: notify.mobile_app_handy
      data: { message: "Bitte lüften!" }
```

Prompt: „Erinnere ans Lüften, wenn CO_2 5 Minuten über 1.000 ppm und das Fenster zu ist.“ – den Code schreibt die KI.

Foto: Pexels

TERMINE

Unser Fahrplan 2026/27

Ort: **[Raum]** · Zeit: **Samstag, [Uhrzeit]**

Sa · Termin 1	**17.10.2026**	Kick-off & fertige Geräte KI-Assistent einrichten, Home Assistant installieren, IKEA, Shelly & Co. einbinden
Sa · Termin 2	**28.11.2026**	Funk & Protokolle Zigbee, Thread, Matter, MQTT
Sa · Termin 3	**06.02.2027**	ESP32 & ESPHome eigene Sensoren und Aktoren bauen
Termin 4	**n. n.**	Automationen & Dashboards Auslöser, Bedingungen, Aktionen – das Haus handelt selbst
Termin 5	**n. n.**	Eigene Projekte eure Idee umsetzen und vorstellen

Foto: Pexels

MITMACHEN

Bist du dabei?

QR-Code scannen oder Link öffnen und das Formular ausfüllen.

tinyurl.com/iot-htl-leonding

Freigegenstand IoT · HTL Leonding · 2026/27

Foto: Pexels