



AFRALTI

IoT FOR PERSONAL DEVELOPMENT,

A DIY Connected Home Tutorial – Summary

AFRALTI
YOUR PARTNER IN ICT KNOWLEDGE

REVISION 1.0, 2020

OUTLINE

- DIY Connected Home, Background
 - ▶ The Connected Home, An Overview
 - ▶ DIY Connected Home Tutorial, Objectives
- DIY Connected Home, Devices
 - ▶ Generic IoT System Components
- DIY Connected Home, Systems
 - ▶ Open Source Home Automation
 - ▶ Internet Technologies
- DIY Connected Home, Projects
 - ▶ A DIY Connected Home Lighting Solution
 - ▶ A DIY Connected Home CCTV Solution
- Appendix
 - ▶ Acronyms and Naming Conventions
 - ▶ What Next?



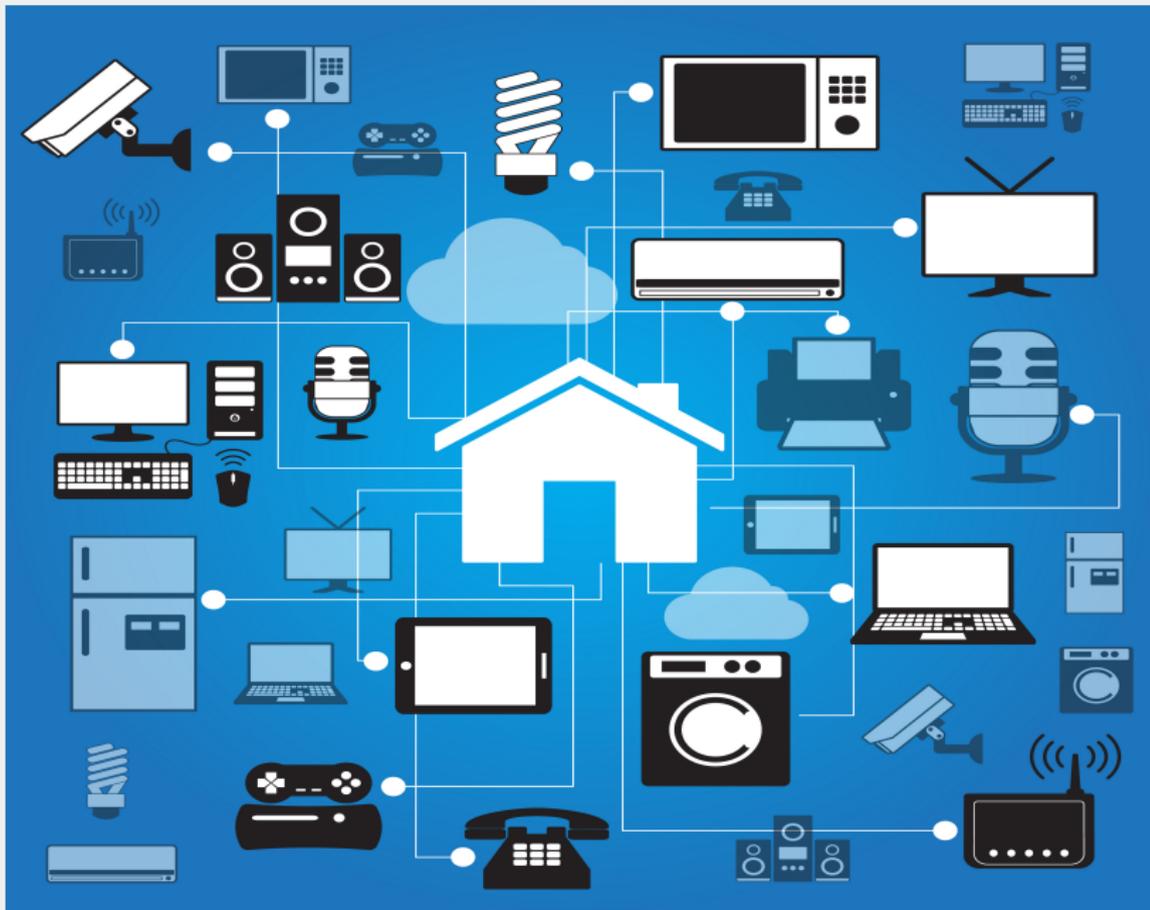


DIY CH, BACKGROUND

THE CONNECTED HOME



AFRALTI



THE CONNECTED HOME, AN OVERVIEW



AFRALTI



A Basic CH Setup

THE CONNECTED HOME, AN OVERVIEW

■ Connected Home

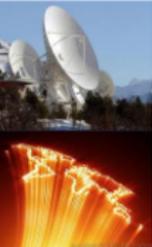
- ▶ Individual appliances are IoT capable and may be smart too (e.g. support AI functions such as voice or gesture)
- ▶ Human element explicitly issues commands or configures automation; typically involves a central IoT G/W and Hub
- ▶ Example: [AFRALTI Connected Home Tutorial](#)

■ Smart Home

- ▶ Next generation CH with central AI running entire home autonomously – a home with a “mind” of its own
- ▶ Learns from inhabitant’s habits/preferences; unburdens them from making repeated or explicit commands for mundane operations
- ▶ Example: Tony Stark and Jarvis



DIY CH TUTORIAL OBJECTIVES



AFRALTI



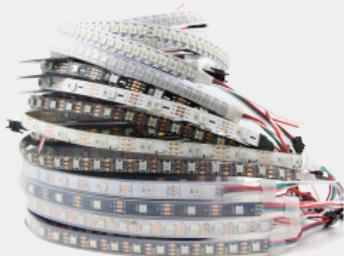
DIY CH TUTORIAL OBJECTIVES, DEVICE-LEVEL

- Demonstrate how to build DIY versions of ...



DIY CH TUTORIAL OBJECTIVES, DEVICE-LEVEL

- ... using generic IoT system components such as:



DIY CH TUTORIAL OBJECTIVES, SYSTEMS-LEVEL

- Discuss IoT/Internet technologies and Open Source platforms adaptable to local DIY CH setups
- Harness an Open Source Home Automation Hub for a practical and scalable CH setup
- Understand the role of vertical application domains (Computer Vision, Machine Learning) at the CH Edge



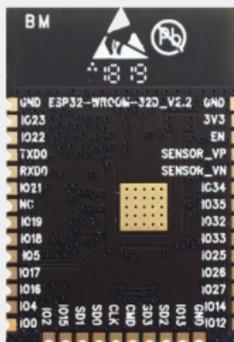
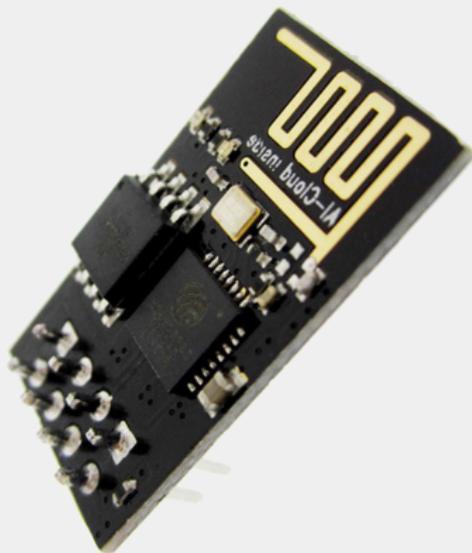


DIY CH, DEVICES

GENERIC IOT SYSTEM COMPONENTS, ESP FAMILY



AFRALTI



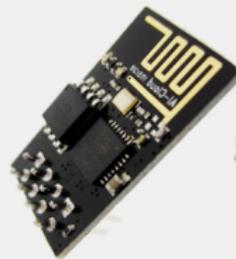
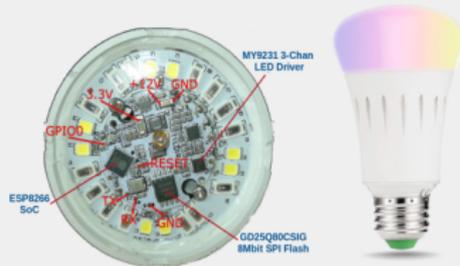
ESP8266 SoC

- Developed by [Espressif](#)
- SoC includes Xtensa-based MCU plus Wi-Fi transceiver
- Requires ESP-NN modules or DEVKIT boards for support components e.g. flash memory, GPIO breakout, antenna
- Very popular; numerous IoT application scenarios
- Native C SDK (Official is by Espressif)
 - ▶ [Espressif ESP-IDF style SDK for v3.0 \(and later\)](#) @ [Espressif ESP8266 RTOS SDK Github](#)
- ESP8266 core for Arduino C/C++ API compatibility
 - ▶ Allows sketch dev for ESP8266 using standard Arduino functions and libraries on Arduino IDE or Platform IO
 - ▶ Open source project by “ESP8266 Community Forum” @ [Community ESP8266 Core for Arduino Github](#)
- Lua, MicroPython, JavaScript, etc support also available



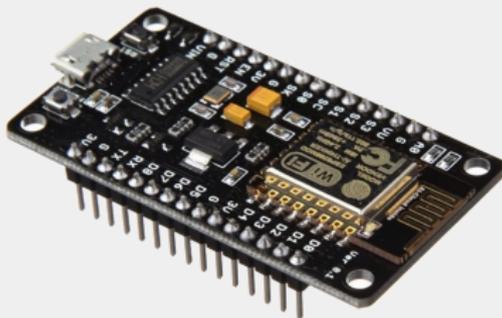
ESP8266 MODULES, ESP-NN

- Minimalistic modules around ESP8266 SoC incorporating
 - ▶ External flash memory, GPIO breakouts, antenna
- Require external USB-to-UART converter for F/W dnwld
- Different ESP-NN have different physical sizes and flash memory, GPIO pin breakouts, etc
- See [▶ ESP8266 Modules](#)
- CH device application examples
 - ▶ Lohas Smart Bulbs
 - ▶ ESP-01/01s WiFi Relay



ESP8266 DEVKITS

- Include several features around ESP8266 SoC or ESP-NN module to facilitate firmware development
 - ▶ NodeMCU, Sparkfun 8266 Thing, Adafruit HUZZAH ESP8266, WeMos D1, Olimex MOD-WIFI-ESP8266
- NodeMCU 1.0
 - ▶ Affordable, fully integrated ESP8266 solution
 - ▶ ESP-12E module, 4 MB SPI flash
 - ▶ CP2102 USB-to-UART adapter for direct F/W flashing
 - ▶ NCP1117 voltage regulator, blue LED on GPIO16, and 220k/100k Ohm voltage divider on the ADC input pin



ESP32 SoC

- Xtensa single-/dual-core 32-bit LX6 processor
- WiFi (802.11 b/g/n) and Bluetooth (BT/BLE Dual Mode)
- Multiple power modes and dynamic power scaling
- Advanced Peripheral Interfaces
 - ▶ Touch/Hall, ADC/DAC, PWM, I²C/I²S/SPI, CAN, IR
- Security Features
 - ▶ Secure boot, Flash encryption, Crypto H/W accel (AES, SHA-2, RSA, ECC, RNG)
- ESP32 SoC series
 - ▶ ESP32-DoWD, ESP32-D2WD, ESP32-SoWD, etc
- Integrated ESP32 Modules/DevKits for support compnts
 - ▶ External flash memory, GPIO breakout, antenna, etc
- SDKs
 - ▶ Official Espressif IDF for ESP32 and ESP32-S @ [Espressif ESP32 IoT Development Framework Github](#)
 - ▶ ESP32 Core for Arduino (also by Espressif) @ [Espressif ESP32 Core for Arduino Github](#)



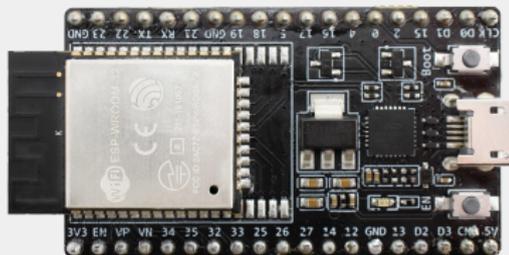
ESP32 MODULES

- Integrate
 - ▶ Crystal oscillator, antenna matching cct, flash mem, etc
- Require
 - ▶ Xternal prog. interfce, bootstrap' resistors, pin hdrs, etc
- ESP32-WROOM-32[D|U]
 - ▶ ESP32-DoWD[Q6] SoC, 4/8/16Mb flash
- ESP32-WROVER[-B|-IB]
 - ▶ ESP32-DoWD[Q6] SoC, 4/8/16MB flash, 8MB PSRAM
- ESP32-SOLO-1
 - ▶ ESP32-SoWD SoC, 4MB flash
- ESP32-PICO-D4 SiP module integrates
 - ▶ ESP32 SoC, crystal oscillator, RF matching circuit, filter capacitors, 4MB flash mem



ESP32 DEVKITS

- ESP32-PICO-KIT V4.1 (Smallest ESP32 DevKit)
 - ▶ ESP32-PICO-D4 module
 - ▶ External antenna, LDO, USB-UART bridge
- ESP32 DevKitC V4
 - ▶ ESP32-WROOM-32 (or even ESP32-WROVER series)
 - ▶ External antenna, USB-to-UART bridge



- ESP-WROVER-KIT V4.1
 - ▶ ESP-WROVER-B module (or other ESP module)
 - ▶ Dual port USB-to-UART and JTAG interface
 - ▶ MicroSD card slot, 3.2" SPI LCD screen, cam module hdr
- ... and many more variants e.g. the ESP32 TTGO T-Beam DevKit integrating LoRa and GPS radios





ESP32 CAMERAS



ESP32-Cam



ESP-EYE



T-Journal



TTIGO T-Plus



TTIGO T-Cam



M5 ESP32-Cam

- Camera: 2MP OV2640 color sensor
- PSRAM: 4MB (ESP32-Cam), 8MB (ESP-EYE, TTIGO)
- Microphone: ESP-EYE, TTIGO, M5-Camera (add-on)
- Programmer: ESP-EYE, T-Journal, TTIGO, M5-Camera
- GPIO: ESP32-Cam (10), T-Journal (4), TTIGO-Plus (2)

GENERIC IOT SYSTEM COMPONENTS, RASPBERRY PI



AFRALTI



RASPBERRY PI

- General purpose SBC with array of features and capabilities; Different models (e.g. ZeroW, 3B, 4B)
- H/W platform for numerous innovative applications
- **Home Automation Hub and IoT G/W:** Platform of choice by major O/S Hubs e.g. [▶ Home Assistant](#) and [▶ OpenHAB](#)
- Entertainment and Information Display Systems
 - ▶ Home Theater or Media Center, Set-top box or Smart TV platform, Digital Video Recorder, Digital Signage Player
 - ▶ Media Player: [▶ Kodi](#) (formerly XBMC)
 - ▶ JeOS: [▶ LibreELEC](#) [▶ OpenELEC](#) [▶ OSMC](#) [▶ Xbian](#) [▶ GeeXboX](#)
- UAV Flight Controller [▶ e.g. Navio2](#) or Ground Control Station
 - [▶ e.g. QGroundControl](#)
- Popular H/W platform for [▶ Robot OS](#)
- Huge and vibrant on-line community and forums, e.g.
 - ▶ [▶ Official Raspberry Pi Forums](#)
 - ▶ [▶ Raspberry Pi Stack Exchange Q/A](#)
 - ▶ [▶ Raspberry Pi Github Community](#)



RASPBERRY PI, MODEL 4B FEATURES



Broadcom 2711
4 Core Cortex A72 @ 1.5GHz
VideoCore VI
H.265, H.264, OpenGL ES 3.0

2.4/5GHz WiFi
BlueTooth 5.0

Micro SD Card
Slot (UnderSide)

2-lane MIPI DSI
Display port

USB-C Power
Port 5V/3A

40 pin GPIO
header

SDRAM
1, 2, or 4GB

PoE HAT Hdr

Gigabit
Ethernet

2 x USB 3.0

2 x USB 2.0

2 x micro HDMI
1x HDMI 4Kp60
or
2x HDMI 4Kp30

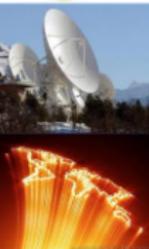
2-lane MIPI CSI
Camera port

4-pole stereo
audio



RASPBERRY PI, DIY CH GATEWAY

- Benefits from superb IoT stacks on GNU/Linux ecosystem
 - ▶ M2M: MQTT (Mosquitto, RabbitMQ, Eclipse Paho, etc)
 - ▶ Web: Langs (Node.JS, Python, ...), Proxy (Caddy, Nginx, ...)
 - ▶ Cloud Native: Docker, Git, Snap, etc
 - ▶ HA Hub: Home Assistant, OpenHAB, Domoticz, etc
 - ▶ IoT Tools: Node-RED, InfluxDB, Grafana, etc
 - ▶ Misc.: VPN (e.g. PiVPN), Email (e.g. ssmtps), DDNS
- Supports variety of wireless H/W interfaces
 - ▶ On-board: 2.4/5GHz WiFi, Bluetooth 5.0
 - ▶ USB/GPIO: 802.15.4, RFID/NFC, ZigBee, Cellular (3G/4G/5G)
- Enjoys misc. Linux support for heterogenous networking transports and communication protocols at the CH edge
 - ▶ 802.15.4: PHY and MAC with (optional) 6LoWPAN stack
 - ▶ 6LoWPAN over BLE
 - ▶ 802.11s Mesh, e.g. BATMAN(-adv), OLSR
 - ▶ See also [Linux Networking Documentation](#)



RASPBERRY PI, DIY CH GATEWAY



A Raspberry Pi Home Automation Hub and IoT G/W Setup





DIY CH, SYSTEMS

OPEN SOURCE HOME AUTOMATION



AFRALTI



Image courtesy of [CCOSTAN](#)

HOME AUTOMATION HUBS

- Servers acting as a central points of communication btm smart devices for practical adoption of CH systems
- Export a single app/dashboard interface
 - ▶ Hundreds of smart home devices each with unique app – even for devices from same vendor
 - ▶ Central hub provides uniform interface to all integrated devices for smart control from phone/tablet/laptop
- Manage interactions btm diverse automation tasks
 - ▶ Export consistent automation config rules for all integrated devices
 - ▶ Allow activation of devices/systems upon notifications from unrelated devices/systems
- Open Source options present a single solution platform
 - ▶ Vendor/technology agnostic; Generic vs. Vendor lock-in
 - ▶ Flexible, extensible; allow integration of different vendor or arbitrary DIY devices/systems



HOME AUTOMATION HUBS, OPTIONS

■ Popular Commercial Solutions



■ Notable Open Source (DIY) Alternatives



▶ [Home Assistant Website](#)

▶ [OpenHAB Website](#)

▶ [Domoticz Website](#)

▶ [Home Assistant Github](#)

▶ [OpenHAB Github](#)

▶ [Domoticz Github](#)

O/S HOME AUTOMATION, HOME ASSISTANT

- Open Source ▶ Home Automation Platform written in Py3K
- Privacy by design – not dependent on cloud
- Hub platforms
 - ▶ RPi 3/3+/4, Intel NUC i3/i5/i7/i9, Laptop/Desktop, etc
 - ▶ VirtualBox or QEMU machine emulator, Docker container
- Large ▶ Hass Github Community @ 2K contributors, 33K+ stars!
- Modular, extensible via ▶ integrations, >1600 ready-made
 - ▶ Popular IoT devices/services e.g. Philips Hue, Google Assistant, Amazon Alexa, IFTTT, ESPHome, SmartThings
 - ▶ M2M/IoT protocols/transport e.g. MQTT, Zigbee, Z-Wave
- Configuration via YAML files
- Authentication via user profile w/ MFA option
- Automatic discovery of devices



INTERNET TECHNOLOGIES



AFRALTI

HTML



JS



CSS



Let's Encrypt

NGINX



WebRTC

webm

opus

git

INTERNET TECHNOLOGIES

- Communication standards and Internet application platforms for connected home devices/systems
- Open communication protocols/formats facilitate interoperability btm devices
 - ▶ MQTT, HTTP, WebSockets, TLS, RTP, RTSP, etc
 - ▶ 802.11, 802.15.4, BLE, etc
 - ▶ Data Interchange formats: JSON, YAML, CSV, etc
 - ▶ Internet streaming media codecs/containers
- Open Source application frameworks/libraries facilitate development and integration of sub-systems
 - ▶ Servers: MQTT (Mosquitto,...), Web (Caddy, NGINX,...)
 - ▶ Lang. web frameworks (Node.js, Python,...), containers (Docker,...), and deployment tools (Git) for microservices
 - ▶ Data Analytics/Visualization (Grafana, InfluxDB, ...)
 - ▶ etc
- Infrastructure for OpenSource home automation systems



INTERNET TECHNOLOGIES, TOPICS

■ M2M with MQTT

- ▶ MQTT Model: Client/Broker/Bridge, Publish/Subscribe
- ▶ MQTT Features: QoS, Keep alive, Session, Retain, Wills

■ Web

- ▶ HTML5: HTML, CSS, JavaScript
- ▶ Web API, Web Frameworks, Cloud Native
- ▶ HTTP, WebSockets

■ Security Layers

- ▶ TLS, Let's Encrypt

■ SmartPhone App Technologies

- ▶ Native, Web and Hybrid Apps



INTERNET TECHNOLOGIES, TOPICS

■ Media Streaming

▶ Protocols

- RTP, RTSP, RTMP, HTTP-based (HLS, MPEG DASH)

▶ Codecs

- Audio e.g. AAC, Opus, Vorbis
- Video e.g. H.264, H.265, AV1

▶ Containers

- MPEG-2 TS, WebM, Fragmented MP4, etc

▶ IoT Live Streaming

■ WebRTC

▶ Overview of WebRTC Architecture

▶ WebRTC API (getUserMedia, RTCPeerConnection, RTCDataChannel)

▶ WebRTC signaling and media streaming protocols

▶ WebRTC Security

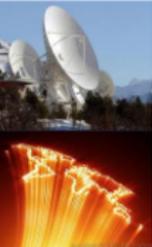
▶ WebRTC P2P and NAT Traversal (STUN, TURN, ICE)





DIY CH, PROJECTS

CONNECTED HOME LIGHTING



AFRALTI



Image courtesy of Forbes Media LLC.

A DIY CH LIGHTING SOLUTION

■ Project Objectives

- ▶ DIY IoT lighting w/ a Home Automation Hub config for scalable light unit installations
- ▶ Commodity H/W and Open source S/W components

■ Desirable System Features

- ▶ Intuitive and hub-managed control interface
 - light unit selection, hue, intensity, effects, automation, etc
- ▶ IoT device/system security
 - user access authentication
 - device secure boot and flash encryption (ESP32)
- ▶ Remote cloud access (if desired)



A DIY CH LIGHTING SOLUTION, COMPONENTS

■ DIY IoT Light Unit

- ▶ LED strip: WS281x, SK6812, SK9822, APA102, etc
- ▶ DC PSU, e.g. 5V/60mA @ RGB; 8A/40W @ 130-150 LEDs
- ▶ Compute/WiFi module: ESP8266/ESP32, RPi Zero-W, etc

■ DIY User Interface

- ▶ Web or Smartphone app
- ▶ AI (Voice, Gesture)

■ DIY Home Automation Hub Integration

- ▶ S/W: Home Assistant, OpenHab, etc
- ▶ H/W: RPi 3B or 4B, Intel NUC etc



A DIY CH LIGHTING SOLUTION, HASS INTEGRATION

- Variety of Hass DIY integration options for ESP-based light controllers e.g.
 - ▶ Integration via [Hass MQTT Light](#)
 - ▶ Integration via [Hass ESPHome](#)
 - ▶ Integration via [Hass WLED](#)
- DIY integrations allow
 - ▶ flexible control via MQTT or REST API
 - ▶ support for std functions (e.g. ON/OFF, brightness, RGB) as well as custom LED strip lighting effects
- Related ESP DIY Frameworks
 - ▶ [ESPHome](#) allows control of ESP8266/ESP32 devices via config files and Home Automation systems
 - ▶ [Tasmota](#) Open source firmware for ESP8266 devices



CONNECTED HOME CCTV



AFRALTI



A TYPICAL CH CCTV SETUP



AFRALTI



Indoor (living room)
F19828P/9821P



Garage
F19805W/9803/9903



FosBaby
Baby room



Corridor
F19851P



Image courtesy of 



A DIY CH CCTV SOLUTION

■ Project Objectives

- ▶ Scalable DIY Home surveillance IP-cam CCTV system
- ▶ Commodity H/W and Open source S/W components
- ▶ User privacy and control



■ Desirable System Features

- ▶ Low latency streaming using end-to-end secure media transport protocols
- ▶ User access authentication
- ▶ Globally/WAN accessible media stream on ubiquitous GUI (e.g. HTML5 web browser)
- ▶ GUI features e.g. stream recording, camera selection in multi-cam setups, etc
- ▶ IoT interfaces
 - Remote actuator control (e.g. Camera PTZ, relay device)
 - Silent alarm integration: Motion sensing (e.g. via PIR)
 - Smart silent alarm integration: AI (CV/ML) edge computing

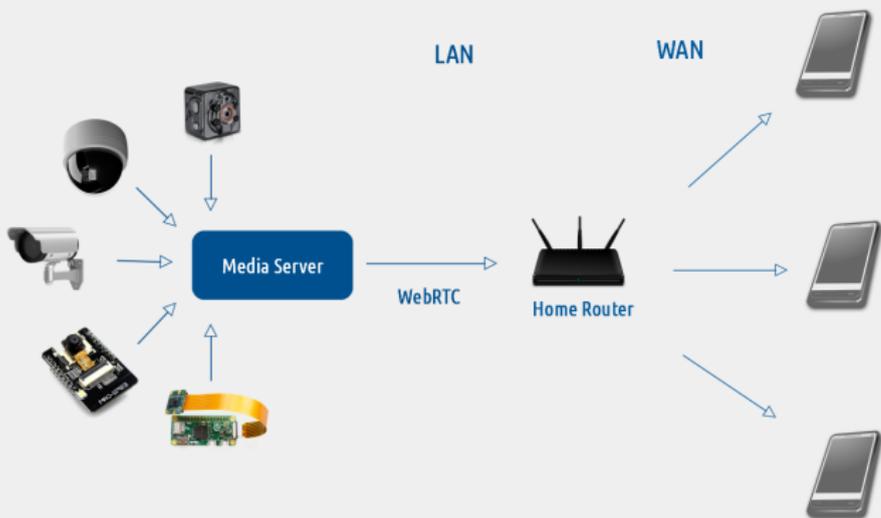


A DIY CH CCTV SOLUTION, COMPONENTS

- Off-the-shelf H/W e.g. RPi, Intel NUC, Jetson Nano/TX
 - ▶ Native interfaces for “embeddable” cams (e.g. RPiCam, Logitech-CXXX USB Cams, ESP32-Cam)
 - ▶ VPU/GPU for realtime video analytics (Movidius USB module for Raspberry Pi)
 - ▶ HDMI, MIPI DSI, etc for 2-way voice/video telepresence
 - ▶ GPIO for DIY PTZ camera, relay, solenoid, etc
- Open Source S/W infrastructural support
 - ▶ Embedded GNU/Linux OS
 - ▶ Plenty of OpenSource WebRTC modules @ Github
 - ▶ OpenSource CV and ML frameworks (e.g. OpenCV, Keras, Tensorflow, PyTorch, etc)
 - ▶ Integration with Home Assistant also possible



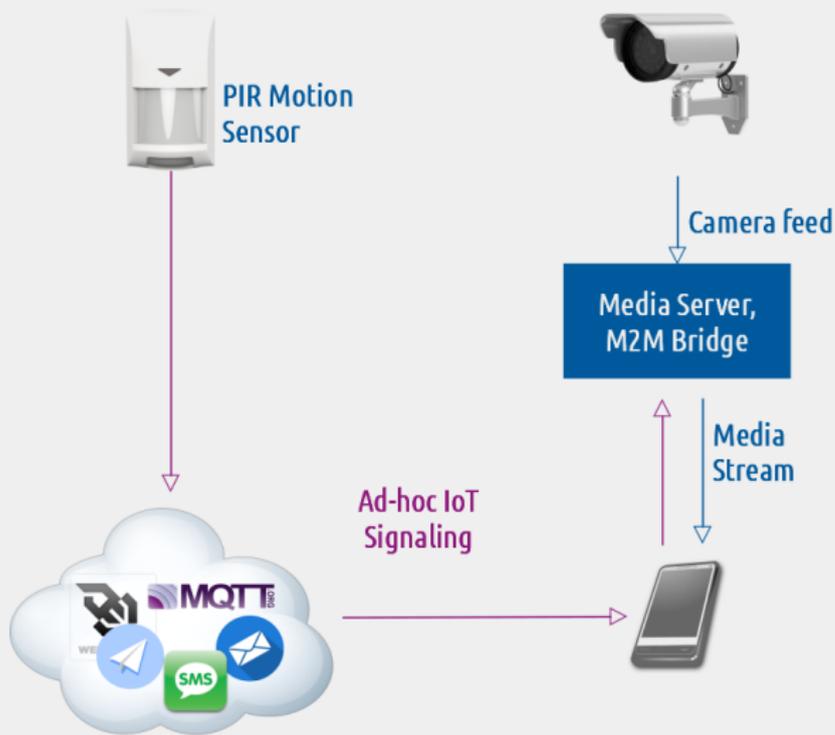
A DIY CH CCTV SOLUTION, BASIC SETUP



- Several (ONVIF) IP-cam options; Commercial or DIY
- Setup tested on Zuku, Airtel, Telkom, and Safaricom ...
- ... with many practical cases requiring only WebRTC P2P/STUN; no 3rd party TURN CSaaS or VPN cloud



A DIY CH CCTV SOLUTION, PIR INTEGRATION

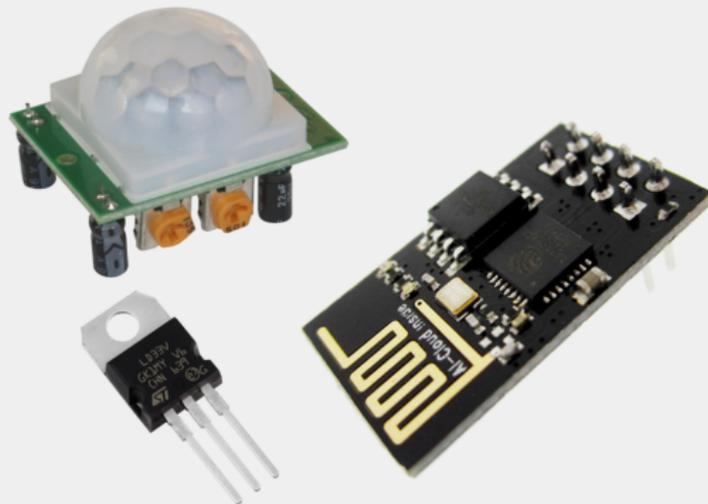


- Basic IoT Architecture for “Dumb” Silent Alarm
- IoT signaling allows both Vertical and Horizontal scaling



A DIY CH CCTV SOLUTION, PIR INTEGRATION

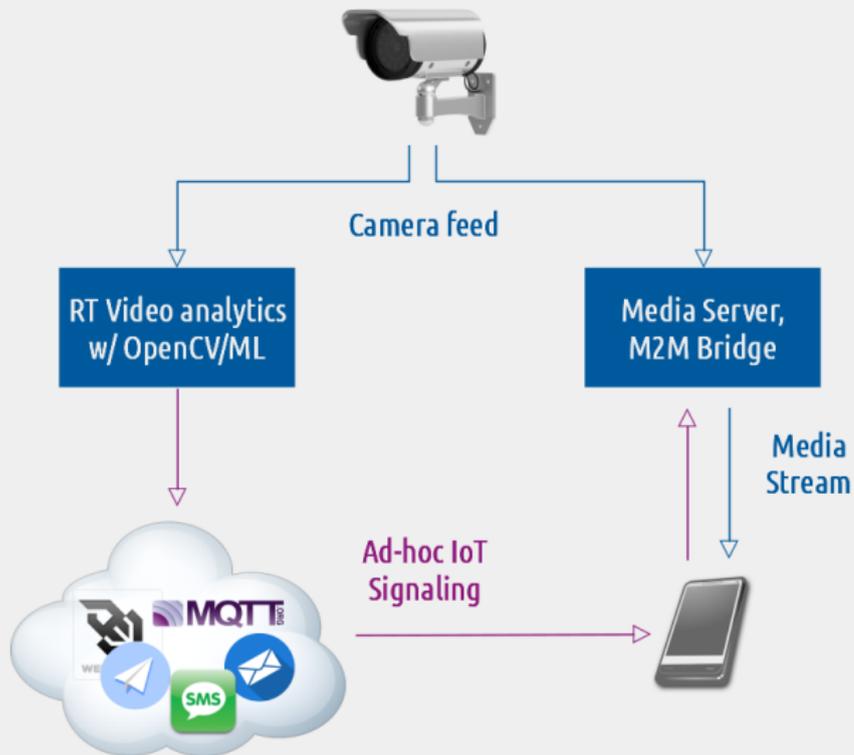
- Key components for DIY IoT PIR motion sensing
 - ▶ HC-SR501 PIR Motion Sensor
 - ▶ ESP8266 ESP-01 Module
 - ▶ LM1117-3.3 (3.3 voltage Regulator)



- Ad-hoc Signaling upon motion detection
 - ▶ SMTP (email), IFTTT (email or SMS), MQTT, WS, etc



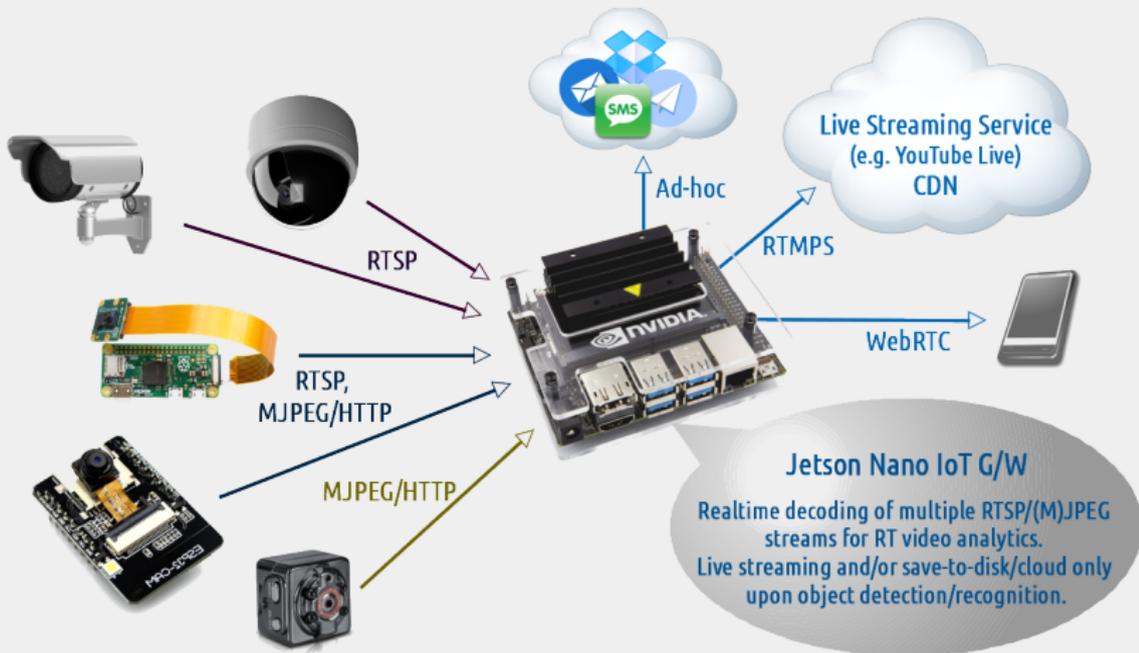
A DIY CH CCTV SOLUTION, AI INTEGRATION



- Basic IoT Architecture for “Smart” Silent Alarm
- IoT signaling allows both Vertical and Horizontal scaling



A DIY CH CCTV SOLUTION, AI INTEGRATION



Horizontal scaling of smart silent alarm system,
VPU/GPUs at CH edge





A DIY CH CCTV SOLUTION, AI INTEGRATION

NVIDIA® Jetson Nano™

- Smallest and lowest powered (as little as 5W) member of Jetson family (Nano, TX, Xavier) for DL inference and CV
- JetPack SDK: Comprehensive S/W stack for AI applic. dev.
- Supports multiple NN in parallel for image classification, object detection, segmentation, speech processing, etc
- Runs several NN from popular O/S ML frameworks
 - ▶ Keras, TensorFlow, mxnet, PyTorch (and Caffe2), Caffe
- Developer Kit
 - ▶ \$99 embedded board (CoM + carrier board)
 - ▶ Dev. Kit carrier board not intended for production ▶ note*
 - ▶ Production carrier boards available from 3rd party vendors ▶ e.g. JN30B





A DIY CH CCTV SOLUTION, AI INTEGRATION

NVIDIA®Jestson Nano™ Compute-on-Module Specs



AFRALTI

PROCESSOR

| | |
|---------------|---|
| CPU | Quad-core ARM A57 @ 1.43 GHz |
| GPU | 128-core Maxwell |
| Memory | 4 GB 64-bit LPDDR4 25.6 GB/s |
| Video Encoder | 4Kp30 (4x) 1080p30 (9x) 720p30, (H.264/H.265) |
| Video Decoder | (2x) 4Kp30 (8x) 1080p30 (18x) 720p30, (H.264/H.265) |

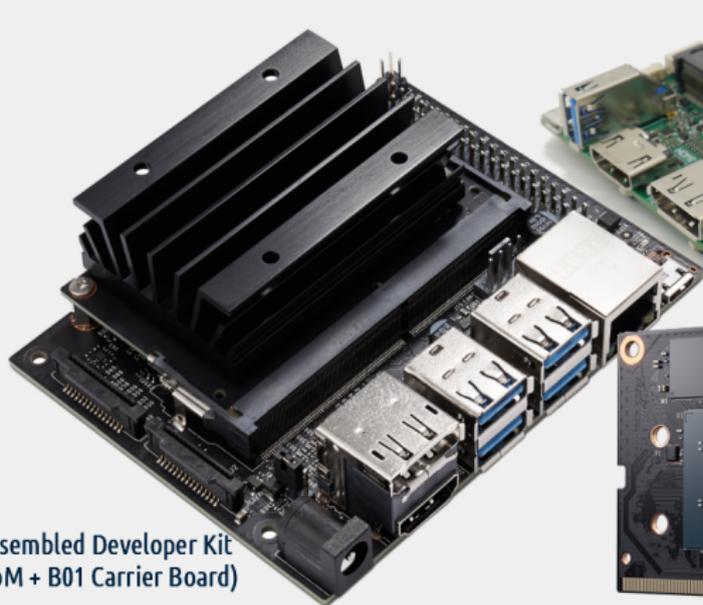
INTERFACES

| | |
|---------------|---|
| USB | 4x USB 3.0, USB 2.0 Micro-B |
| Camera | 2x MIPI CSI-2 DPHY Lanes |
| Display | HDMI Display Port |
| Networking | Gigabit Ethernet (RJ45, PoE) |
| Wireless | M.2 Key-E with PCIe x1 |
| Storage | MicroSD card (16GB UHS-1 Recommended Minimum) |
| 40-pin Header | GPIO I2C I2S SPI UART Audio Clock |
| Power | 5V DC (uUSB, Barrel Jack, PoE) – 5W 10W |
| Size | 80x100mm |



A DIY CH CCTV SOLUTION, AI INTEGRATION

NVIDIA® Jetson Nano™ Embedded Boards



Assembled Developer Kit
(CoM + B01 Carrier Board)



JN30B Carrier Board
Production



Compute-on-Module



APPENDIX

ACRONYMS AND NAMING CONVENTIONS



CH Connected Home

SH Smart Home

HA Home Automation

G/W Gateway

N/W Network(ing)

O/S Open Source

S/W Software

F/W Firmware

H/W Hardware

IoT Internet of Things

M2M Machine-to-Machine

AI Artificial Intelligence

ML Machine Learning

DL Deep Learning

CV Computer Vision

OS Operating System

CE Consumer Electronics

DIY Do it Yourself

WHAT NEXT? SPECIALIZED TUTORIALS FOR CH DIY

■ Mastering Open Source Home Automation

- ▶ Home Assistant
- ▶ Node-Red
- ▶ ESPHome
- ▶ Tasmota

■ Coding and Programming

- ▶ HTML5 (HTML, CSS, JavaScript)
- ▶ Python
- ▶ Embedded C++

■ Exploring ESP8266 and ESP32

- ▶ GPIO, UART, SPI, I2C, I2S, Analog I/O
- ▶ PWM, DMA, Interrupts, Timers
- ▶ MQTT, HTTP, WebSockets, Bluetooth Low Energy



WHAT NEXT? SPECIALIZED TUTORIALS FOR CH DIY

■ Computer Interfacing Electronics

▶ Basic Principles and Circuits

- Voltage, Current, Resistance, Capacitance, Inductance
- Logic Gates
- ADC/DAC, PWM, PCM
- Discrete Components e.g. LEDs, Transistors, Optocouplers

▶ Interfacing w/ sensors and actuators

- LDC/OLED Character Displays
- DC motors, Servos, and Relays
- Temp, Humidity, Hall, and PIR sensors

■ Introduction to Embedded GNU/Linux

- ▶ UNIX Command Line Interface (CLI)
- ▶ Linux System Admin: Linux FHS, SystemD, Udev, etc

■ The Raspberry Pi Platform

- ▶ GPIO, UART, I2C
- ▶ PWM, DMA, Interrupts, Timers
- ▶ WiFi, Bluetooth, ZigBee, Z-Wave
- ▶ Video (Broadcom MMAL, V4L2), Audio (ALSA)





WHAT NEXT? EMBEDDED GNU/LINUX TUTORIALS

Linux Systems Programming

- System calls and Library functions
- UNIX Universal File I/O model
- (POSIX) IPC Mechanisms
 - ▶ Pipes/FIFOs
 - ▶ Message Queues
 - ▶ Shared Memory
 - ▶ Sockets
- Multitasking and Concurrent Programming
 - ▶ Threads, I/O models (signals and events)
 - ▶ IPC synchronization (Semaphores and Mutual exclusion)
 - ▶ Timers and Sleeping
- Virtual Memory
 - ▶ Dynamic memory allocation
 - ▶ Memory mappings
 - ▶ Memory locking
- Realtime Linux Programming



WHAT NEXT? EMBEDDED GNU/LINUX TUTORIALS

Linux Development

- Cross-Development Platforms
 - ▶ Toolchain: Linaro, crosstool-ng
 - ▶ Systems: Busybox, Buildroot
- Boot loaders (U-Boot) CLI and Environment
- Device Tree (Syntax and Bindings)
- Kernel Config, Compile and Booting
- Kbuild System (Makefiles, Kconfig basics)
- Kernel Modules
 - ▶ Compilation, utilities, loading/unloading, symbols
- Device drivers
 - ▶ Device types, Device nodes
 - ▶ File operations and memory mapping
 - ▶ Interrupt handling and DMA
 - ▶ Synchronization primitives

