App.hpptd.cpp

Il codice è un **web server ESP32** che:

* Riceve richieste HTTP per controllare parametri della camera,
* Permette di configurare clock, registri, risoluzione, PLL,
* Espone un’interfaccia web per interagire con la camera,
* Permette di catturare foto o video in streaming,
* Gestisce tutto tramite HTTP GET e query parameters.

Camera\_pins.h

contiene le **definizioni dei pin GPIO per ciascun modello di ESP32-CAM.**

Questo blocco di codice viene attivato **solo se** è definito il macro CAMERA\_MODEL\_AI\_THINKER (di solito tramite #define CAMERA\_MODEL\_AI\_THINKER nel codice). Questo identifica il modello di fotocamera **AI Thinker ESP32-CAM**, uno dei modelli più comuni di moduli camera con ESP32.

**Cosa significa ciascuna riga**

Questi #define associano i pin GPIO dell'ESP32 ai segnali della fotocamera. Ogni segnale fisico della fotocamera è collegato a un pin specifico della CPU ESP32.

* **PWDN\_GPIO\_NUM 32**  
  Pin GPIO 32 controlla il segnale **PWDN (power down)** della fotocamera, usato per spegnere o accendere il modulo camera. Se la fotocamera non ha pin power down, potrebbe essere -1.
* **RESET\_GPIO\_NUM -1**  
  Pin di reset hardware della fotocamera. Qui è -1, significa che non si usa un pin di reset dedicato.
* **XCLK\_GPIO\_NUM 0**  
  Pin GPIO 0 genera il segnale di clock esterno **XCLK** per la fotocamera (tipicamente un clock a 20 MHz).
* **SIOD\_GPIO\_NUM 26** e **SIOC\_GPIO\_NUM 27**  
  Pin 26 (SIOD) e 27 (SIOC) sono i segnali per il bus I²C della fotocamera, usati per configurare il sensore via protocollo SCCB (simile a I²C).

**I pin delle linee dati video (Y9..Y2)**

La fotocamera usa un bus parallelo per i dati video a 8 bit:

* **Y9\_GPIO\_NUM 35**
* **Y8\_GPIO\_NUM 34**
* **Y7\_GPIO\_NUM 39**
* **Y6\_GPIO\_NUM 36**
* **Y5\_GPIO\_NUM 21**
* **Y4\_GPIO\_NUM 19**
* **Y3\_GPIO\_NUM 18**
* **Y2\_GPIO\_NUM 5**

Questi sono i pin GPIO collegati ai 8 bit di dati pixel (Y9 è il bit più significativo, Y2 il meno significativo).

**Segnali di sincronizzazione e clock**

* **VSYNC\_GPIO\_NUM 25**  
  VSYNC indica l’inizio di una nuova immagine (frame sync).
* **HREF\_GPIO\_NUM 23**  
  HREF indica l’inizio di una nuova linea (line sync).
* **PCLK\_GPIO\_NUM 22**  
  PCLK è il clock pixel che scandisce il flusso dati.

**LED di stato**

* **LED\_GPIO\_NUM 4**  
  Questo pin controlla il LED di illuminazione del modulo, per accendere la luce (flash). Di solito il pin 4 accende il LED sulla scheda AI Thinker.

**In sintesi**

Questo blocco mappa ciascun segnale fisico della fotocamera AI Thinker ESP32-CAM su un GPIO specifico dell’ESP32, così il driver software sa esattamente quali pin usare per comunicare e acquisire immagini dal sensore.

Camera\_index.h

**Cos’è generalmente un file camera\_index.h?**

* **È un file header .h che contiene risorse statiche "embedded" nel firmware**.
* In questo caso, è usato per **includere pagine web (HTML/JS/CSS) compresse** come array di byte C nel firmware del microcontrollore.
* Serve per poter **distribuire la pagina web che gestisce o mostra il flusso video della fotocamera direttamente dal firmware**, senza bisogno di un file system esterno.
* In pratica, il file contiene tutto il contenuto del frontend web (la UI che gira nel browser per controllare o visualizzare la camera) già convertito in dati C.
* Il server web embedded nell’ESP32 (o altro MCU) risponde con questi dati quando un client (browser) fa la richiesta.

**Quindi, cosa fa camera\_index.h nel progetto?**

* Fornisce **le pagine web** (spesso un file index.html + altre risorse) **compresse in gzip come array di byte**.
* Permette al firmware di servire queste pagine web in modo molto efficiente, senza accesso a file esterni.
* Viene incluso nel progetto C/C++ e compilato nel firmware.
* Insieme al codice server HTTP, gestisce la visualizzazione della UI che controlla la camera o mostra il video.