

Referencia rápida de la tarjeta HVSupplyBoard V1

(12/2025)

Antes de empezar

La tarjeta está configurada para dar 1800 V a la salida sin más que enchufarla y darle al ON en el interruptor exterior. Por tanto tiene establecido el voltaje de salida manualmente, dando tensión de forma fija. Digamos que es un clon de las fuentes de ICaRO y ORCA en su parte de 1800V.

El ESP32 está pinchado, pero sin alimentación.

Para cambiar lo anterior y trastear, es necesario abrir la caja con mucho cuidado, sobre todo porque los cables de HV van soldados y son cortos (ver foto). Se levanta desde atrás, se quitan los conectores Faston del conector de alimentación (ojo con observar dónde va cada uno), y se deja la tapa delante de la base sin que tiren los cables de HV.



Una vez hecho eso, se pueden quitar y poner jumpers o conectar el USB del ESP32, y volver a poner los faston. Se puede trabajar con la tapa semiabierta aunque es un poco incómodo ...

9	JP7, JP10	JP7 establece si la salida de la fuente de HV se fija en un estado (1-2) o se controla de forma programada (2-3). Si se fija en un estado, JP10 establece el estado: activada (OFF) o desactivada (ON).	1-2, OFF
10	JP11	Establece si el DAC de control se alimenta del raíl AVDD5V (1-2) o 3.3VDD (2-3). (Depuración)	1-2
A		Estado de la salida de HV	
B		Estado de los railes de tensión de AVDD5V, 5VDD y 3.3VDD	
C		Estado del raíl de 12V de entrada	
H1	J9	Header de expansión del bus I2C principal.	
H2	J8	Header de depuración RS232, compatible con cable estándar FTDI USB. NO USAR EL PIN TX SI ESTÁ CONECTADO EL USB DEL ESP32	

Si un estado de un jumper en el prototipo está en **rojo**, no debe ser necesario cambiarlo salvo motivo muy justificado.

Si un estado de un jumper en el prototipo está en **verde**, se puede cambiar con el propósito de avanzar en el desarrollo:

- JP8 y JP7 para pasar de modo manual a programado. Se recomienda NO tocar JP7 y poner JP10 a ON (desactivado) por seguridad mientras se realiza el desarrollo. Para depurar, se puede medir la señal de nivel de HV programada por el DAC en TP10, y la señal de HV ON controlada por el ESP32 en el pin 3 de JP7.
- JP3, una vez completado el desarrollo, se puede poner a ON para alimentar el ESP32 desde la tarjeta. Mientras tanto, si el ESP32 está alimentado por USB, JP3 debe permanecer en OFF.

Más información sobre la tarjeta

La tarjeta se alimenta de una fuente de 12V DC en el conector J1. Este conector tiene además dos terminales para instalar opcionalmente un switch para alimentar o no la tarjeta. Si no se usa, es necesario instalar el jumper JP1, y la placa siempre estará alimentada. Esto permite instalar un interruptor en el frontal de un panel para encender y apagar la fuente.

Tiene 4 railes de alimentación:

- +12V (VDC)
 - o Procedentes de la alimentación externa.
- +5V (5VDD)
 - o Proporcionados por un regulador LM2596 alimentado por VDC (JP2) u opcionalmente por el ESP32 (JP3).
- +5V (AVDD5V)
 - o Proporcionados por un regulador TLV1117 alimentado por VDC, o por el rail 5VDD (JP4).
- +3.3V (3.3VDD)
 - o Proporcionados por un regulador TLV1117 alimentado por el raíl 5VDD (JP5), o por el ESP32 (JP6).

Esto permite, opcionalmente:

- Operar normalmente: el sistema se alimenta de 12V, y el ESP32 del sistema (funcionamiento nominal). JP2 y JP5 ON; JP3, JP4 y JP6 OFF

- Operar sin fuente de 12V externa en algunos casos: se alimenta el ESP32 por USB, y este alimenta todos los railes de 3.3V y 5V. JP2 y JP5 OFF; JP3, JP4 y JP6 ON.
- Operar de manera nominal pudiendo desarrollar sin riesgo con el ESP32 alimentado por USB. JP2 ON; JP3, JP4, JP5 y JP6 OFF

Control de la fuente de HV

La fuente de HV tiene dos controles: encendido y apagado de la salida de HV, y nivel de tensión de salida. La tarjeta permite controlar ambas cosas manualmente, o desde el ESP32.

Encendido y apagado de la salida HV:

- JP7 selecciona entre control de encendido, fijo (1-2) o programado (2-3).
- JP8 selecciona, para el encendido fijo, si la fuente está encendida (jumper abierto) o apagada (jumper cerrado). Hay un led HV ON que indica el estado de la salida.
- Para el control programado, el ESP32 emplea un GPIO (GPIO32).

Nivel de tensión de salida:

- JP8 selecciona entre nivel de salida manual (2-3) o programado (1-2).
- En modo manual el potenciómetro R17 establece el nivel de salida. Girar en sentido horario aumenta la tensión, y en sentido antihorario la disminuye.
- Para el control programado, el ESP32 emplea un DAC de precisión conectado por I2C.

El nivel de salida es proporcional a la tensión que puede medirse en TP10 (HV SET), que varía entre 0V y una tensión de referencia que puede medirse en TP11 (HV REF). En el caso del prototipo esta tensión es de 2.5 V y la fuente tiene un fondo de escala de 2 kV; por lo que para que ofrezca 1.800 V a la salida HV SET está establecido a 2.25 V.

Monitorización de la fuente de HV

Se monitorizan varias cosas de la fuente de HV:

- Mediante un ADC de precisión se puede medir:
 - o La tensión de control HV SET (0-2.5V)
 - o La tensión de monitorización de la salida, HV OUT (0-2.5V)
 - o La corriente consumida por la carga de HV mediante la tensión en un shunt (desactivado por ahora porque tiene su peligro).
 - o La tensión de referencia HV REF que da la propia fuente (2.5V)
 - o La tensión de la fuente del propio chip (que es AVDD) (5.0V)
 - o La temperatura del dispositivo
- Mediante un monitor de potencia se puede medir:
 - o La tensión de alimentación de baja del módulo de HV
 - o La corriente consumida por el módulo de HV

En ambos casos, tanto el ADC como el monitor de potencia se controlan mediante I2C

Elementos conectados al bus I2C

Todos los elementos anteriores están conectados al mismo bus I2C controlado por el ESP32. El canal I2C está en los pines 3 y 6 (GPIO22 y GPIO21).

Los chips concretos y sus direcciones I2C están en la siguiente tabla.

Elemento	Chip concreto	Dirección
DAC	DAC70501ZDGSR	78h (1001 000)
ADC	ADS112C04IPWR	70h (1000 000)
Power monitor	INA226AIDGSR	75h (1000 101)

Identificación

Los chips anteriores esperan diferentes secuencias de bytes en el I2C, tienen similitudes pero son todos diferentes.

DAC

Funcionamiento general:

- 1 byte de dirección donde los bits [7:1] son la dirección y el bit LSB es R/W (0/1).
- 1 byte de comando.
- 2 bytes de datos (MSDB y LSDB), que se leen o escriben según el comando y el bit R/W.

Para identificar el chip hay que enviar el comando DEVID (01h), y tendrá un valor esperado 1115h. Los 4 bytes serían:

Elemento	Byte 1 (addr+rw)	Byte 2 (cmd)	Bytes 3&4 (MSDB&LSDB)
DAC70501ZDGSR	90h	01h	1115h

Power monitor

Funcionamiento general a base de registros de 16 bits que se leen y escriben.

Para escribir un registro, se envía una secuencia de 4 bytes que son:

- 1 byte de dirección donde los bits [7:1] son la dirección y el bit LSB es W (0).
- 1 byte de dirección del registro a escribir.
- 2 bytes de datos, que escriben en el registro.

Para leer un registro, se hace en dos partes:

- Se envía una secuencia de 2 bytes para escribir el registro a leer en la siguiente operación. Estos dos bytes son idénticos a los dos primeros de la operación de escritura.
- Se envía una secuencia de 3 bytes para leer los contenidos del registro:
 - o 1 byte de dirección donde los bits [7:1] son la dirección y el bit LSB es R (1).
 - o 2 bytes de datos, que son el contenido del registro.

En nuestro caso particular, el registro FEh es el *Manufacturer ID Register*, y debe contener 5449h, y el registro FFh es el *Die ID Register*, y debe contener 2260h. Para leer los dos registros se debe enviar la secuencia:

1: 8A FE

2: 8B 54 49 (54 49 sería la respuesta)

3: 8A FF

4: 8B 22 60 (22 60 sería la respuesta)

ADC

Funcionamiento general a base de comandos por un lado, y registros de 8 bits que se leen y escriben por otro (mediante un comando RREG o WREG en cada caso). En todos los casos, el primer byte contiene la dirección del esclavo y el LSB indica lectura (1) o escritura (0), como en los anteriores.

Para leer un registro, se hace en dos partes:

- Se envía una secuencia de 2 bytes para indicar el registro a leer en la siguiente operación:
 - o 1 byte de dirección donde los bits [7:1] son la dirección y el bit LSB es W (0).
 - o 1 byte de con el comando RREG, que lleva incorporado el número de registro a leer y es 0010 rrrx (donde rr es el número de registro, y xx es don't care). Por ejemplo, para leer el registro 2 el comando sería 001010xx (por ejemplo, 28h).
- Se envía una secuencia de 2 bytes para leer los contenidos del registro:
 - o 1 byte de dirección donde los bits [7:1] son la dirección y el bit LSB es R (1).

- 1 bytes de datos, que son el contenido del registro.

No existe un registro o comando de identificación, pero sí se puede leer cualquiera de los 4 registros, que tras un reset son todos 00h. La secuencia para hacerlo por tanto podría ser:

- 1: 80 20
- 2: 81 00 (00 sería la respuesta)
- 3: 80 24
- 4: 81 00 (00 sería la respuesta)
- 5: 80 28
- 6: 81 00 (00 sería la respuesta)
- 7: 80 2C
- 8: 81 00 (00 sería la respuesta)

Trabajo con cada chip

DAC

TBD

ADC

TBD

Power Monitor

TBD

Otras facilidades de la tarjeta

Conector UART

Se encuentra en J8 (H2 en la figura). Está directamente conectado a los pines RXD y TXD del ESP32. Por tanto, puede conectarse a él un cable USB-RS232 3.3V típico, como el de la BeagleBone, para depuración.

Es importante no conectar este cable cuando el ESP32 está alimentado por USB, ya que este controla las líneas RXD y TXD. En este caso, como máximo puede usarse la línea RXD para monitorizar la salida.

Conector I2C

Se encuentra en J9 (H1 en la figura). Es directamente el mismo bus I2C donde están conectado el resto de elementos, y permite conectar lo que se quiera; desde un sensor de presión/temperatura a un display. Para depuración con un osciloscopio hay unos test points cercanos.

Otros elementos y headers no comentados

La tarjeta tiene otras facilidades que de momento no se han comentado pero que se podrían usar si se quiere:

- 4 pushbuttons conectados a varios GPIOs del ESP32.
- Un encoder con pulsador como el que se puede encontrar en la tarjeta de Espressif ESP32-C3-LCDKit (no puede usarse al mismo tiempo que 3 de los pushbuttons anteriores)
- Un header de 8x2 multiuso donde se exportan varias fuentes de alimentación y GNDs, además de varios pines del ESP32 compatibles con GPIOs y varios canales SPI e I2C.
- Un header para un LCD SPI bastante estándar (ST7789V).