Logic Level Converter

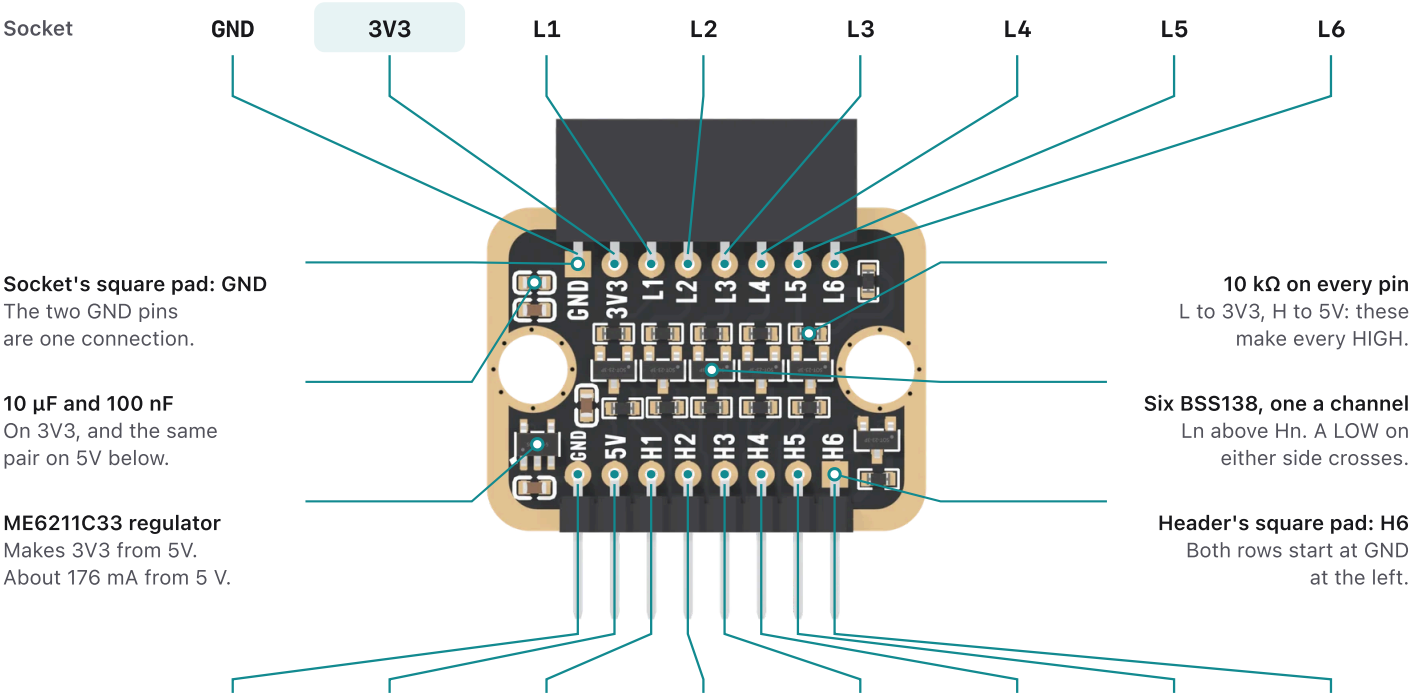
Six channels between 3.3 V and 5 V. It needs 5V and GND, nothing else.

Never feed the socket's 3V3 pin

It is the output of the board's own regulator. **On a 3.3 V board, wire its 5V (VBUS on a Pico) to the header's 5V and leave its 3V3 pin unwired.** A 3.3 V board's pins go on L1 to L6, never on H.

Pins, and which to wire

Socket at the top: the 3.3 V side, GND, 3V3 and L1 to L6. Header at the bottom: the 5 V side, GND, 5V and H1 to H6. Each L sits above the H of the same number; the two GND pins are one.



Header	GND	5V	H1	H2	H3	H4	H5	H6
I2C block	GND	5V	SDA	SCL	·	·	·	·
TK89 display	GND	5V	CS	MOSI	SCL	DC	RST	BL

5V in, 3.3 V out

The ME6211C33 makes the 3.3 V side from 5V. Without 5V neither side works. A 3.3 V block in the socket runs on it, up to about 176 mA: the 500 mA on the part's sheet assumes a smaller drop than 5 V to 3.3 V.

How fast, and how much

3V3 pin gives

176 mA from 5 V, worked out

The package's 300 mW. An Uno's own 3.3V pin gives 50.

5 V side drives

48 ns to a 3.3 V HIGH, modelled

SPI at about 5 MHz on short wires. An Uno into a TK89.

3.3 V side drives

93 ns to a 5 V HIGH, modelled

SPI to about 2.7 MHz. Serial and I2C always fit.

A LOW crosses at once, either way. A HIGH is finished by a 10 kΩ, so fast clocks and long wires are the limit: slow the clock first.

Currents are the ME6211C33M5G-N sheet's (V14): 300 mW for the SOT-23-5 package, read as room temperature, and 2 mA for the board with all six L lines held LOW. Edges are modelled with KUU's BSS138 threshold at 1.2 V (0.8 to 1.6 V on the sheet) and 25 pF of short jumper wire. Not measured. The sketch compiles for an ESP32-S3, an ESP32 and an Uno.

Loopback, two channels both ways

Arduino IDE · Board: ESP32S3 Dev Module · USB CDC On Boot: Enabled · No library

```
// L1 and L2: Uno 2, 3 (on H1, H2). ESP32 25, 26. S3 4, 5.
// Pico 14, 15. Jumper H1 to H2 (Uno: L1 to L2).
const int OUT_PIN = 4;
const int BACK_PIN = 5;

int sendAndRead(int level) {
  digitalWrite(OUT_PIN, level);
  delay(1);
  return digitalRead(BACK_PIN);
}

void setup() {
  Serial.begin(115200);
  pinMode(OUT_PIN, OUTPUT);
  // No pull-up here: the TK97's own 10 kOhm holds the line.
  pinMode(BACK_PIN, INPUT);
}

void loop() {
  int high = sendAndRead(HIGH);
  int low = sendAndRead(LOW);
  if (high == HIGH && low == LOW) {
    Serial.println("HIGH came back HIGH, LOW came back LOW: ok");
  } else if (high == HIGH) {
    Serial.println("always HIGH: check the jumper, and BACK_PIN");
  } else if (low == LOW) {
    Serial.println("always LOW: check 5V and GND, then OUT_PIN");
  } else {
    Serial.println("backwards: check every wire against the list");
  }
  delay(1000);
}
```

```
HIGH came back HIGH, LOW came back LOW: ok
```

Serial Monitor at 115200, once a second. Always HIGH is the jumper; always LOW is 5V or GND.

Wiring

	ROW	S3	ESP32	PICO	UNO
GND	either	GND	GND	GND	GND
5V	header	5V	5V	VBUS	5V
3V3	socket	nothing	nothing	nothing	nothing
L1	socket	4	25	GP14	jumper to L2
L2	socket	5	26	GP15	jumper to L1
H1	header	jumper to H2	jumper to H2	jumper to H2	D2
H2	header	jumper to H1	jumper to H1	jumper to H1	D3

A 3.3 V board goes on the socket's side with jumper wires; an Uno goes on the header. 5V is the TK97's only supply, VBUS on a Pico. **3V3 is never wired from your board.** The ESP32 columns are GPIO numbers.

Two builds, one rule

Uno and a 3.3 V I2C block: the block's GND, VCC, SDA, SCL plug into the socket's GND, 3V3, L1, L2; the Uno's 5V and GND to the header, A4 to H1, A5 to H2. **Uno and the TK89 display:** the TK89 fills the socket; D10 to H1, D11 to H2, D13 to H3, D9 to H4, D8 to H5, D7 to H6. **A 3.3 V board and a 5 V part:** the board on L, the part on H, the part's VCC on the header's 5V.

When it does not work

Nothing crosses either way

Measure 5V to GND on the header. No 5V, no 3V3, no channel.

Readings right, no signal

Ln must face Hn: L1 with H2 are two channels.

Regulator hot

Too much on 3V3, or a short. Keep under about 176 mA.

A 3.3 V pin reads over 3.3 V

It is on an H pin. Move it to L1 to L6 at once.

Fast SPI fails, slow works

Slow the clock, to 4 MHz or less; shorten the wires.

No I2C address answers

5V and GND first, then SDA on H1 and SCL on H2.

Specifications

Channels	6, bidirectional, no direction pin	**Per channel**	BSS138, 10 kΩ to each side
Supply	5V and GND only	**3V3 pin**	Output, 3.3 V ±2 %
3V3 current	About 176 mA from 5 V	**Low side**	3.3 V, fixed by the regulator
Connectors	8-way socket, 8-pin header	**Board**	30.4 × 22.4 mm