DRV8833 Module

simple-circuit.com

Front View

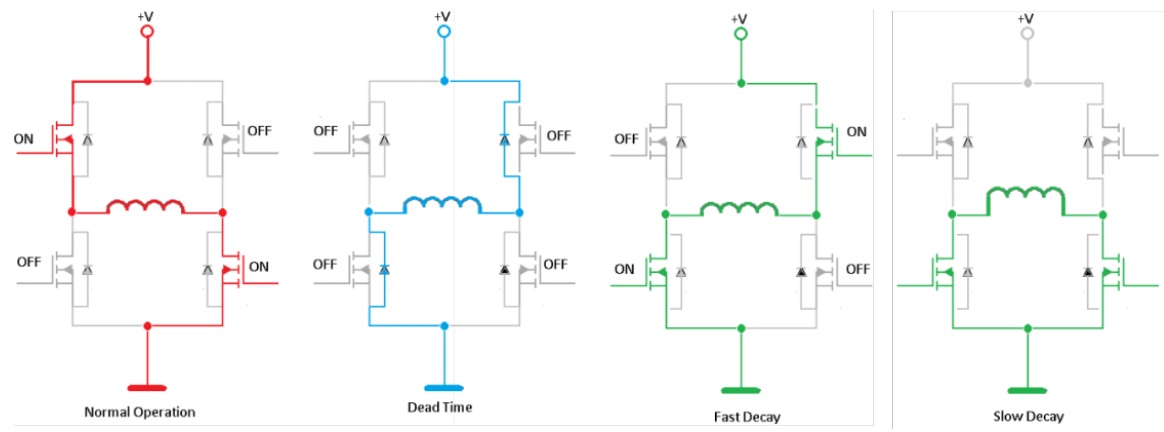
Back View

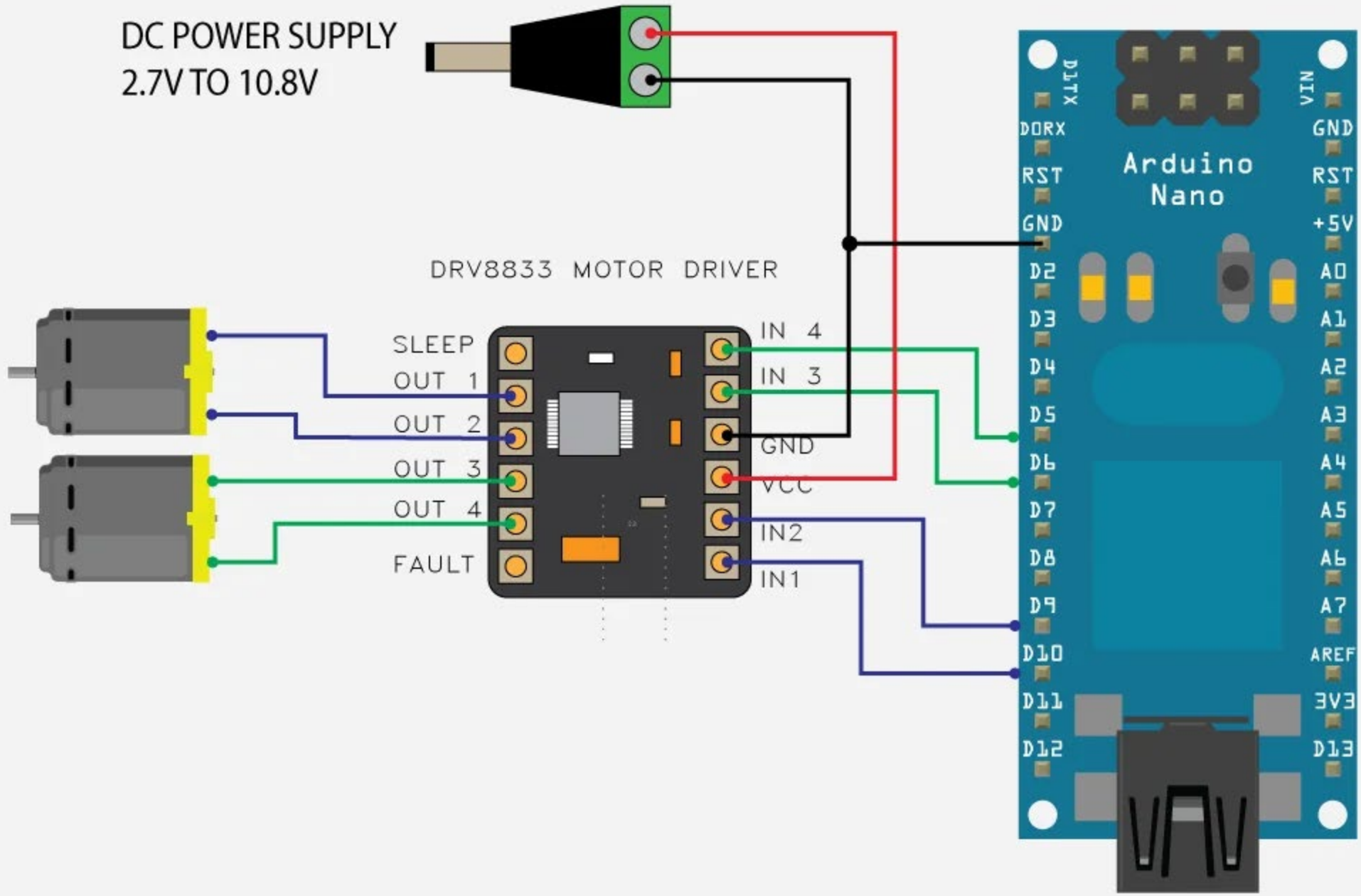
Table 1. H-Bridge Logic

xIN1	xIN2	xOUT1	xOUT2	FUNCTION
0	0	Z	Z	Coast/fast decay
0	1	L	H	Reverse
1	0	H	L	Forward
1	1	L	L	Brake/slow decay

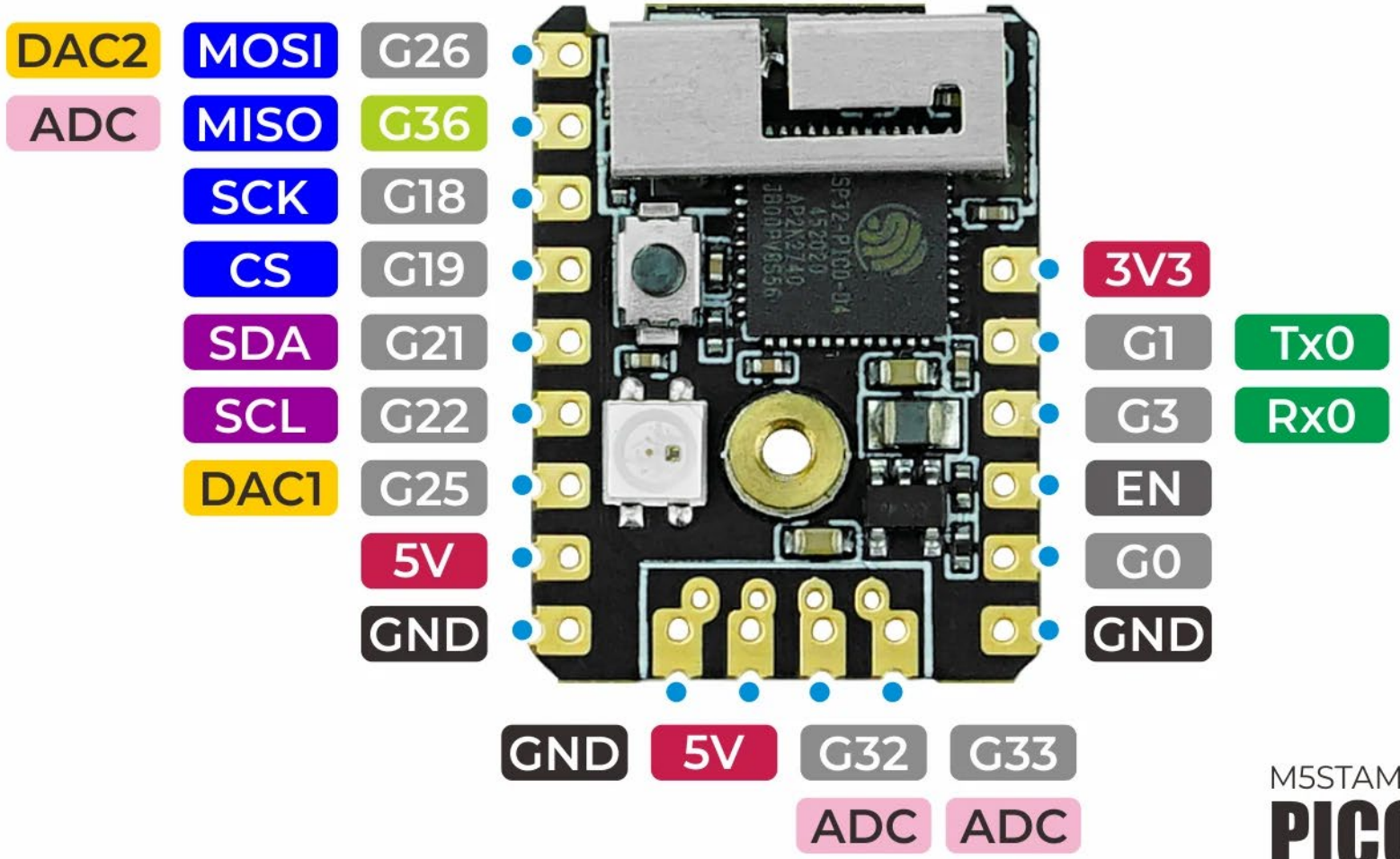
Table 2. PWM Control of Motor Speed

xIN1	xIN2	FUNCTION
PWM	0	Forward PWM, fast decay
1	PWM	Forward PWM, slow decay
0	PWM	Reverse PWM, fast decay
PWM	1	Reverse PWM, slow decay




DC POWER SUPPLY
2.7V TO 10.8V
DRV8833 MOTOR DRIVER
SLEEP
OUT 1
OUT 2
OUT 3
OUT 4
FAULT
IN 4
IN 3
GND
VCC
IN2
IN1
Arduino Nano
D1TX
D0RX
RST
GND
D2
D3
D4
D5
D6
D7
D8
D9
D10
D11
D12
VIN
GND
RST
+5V
A0
A1
A2
A3
A4
A5
A6
A7
AREF
3V3
D13

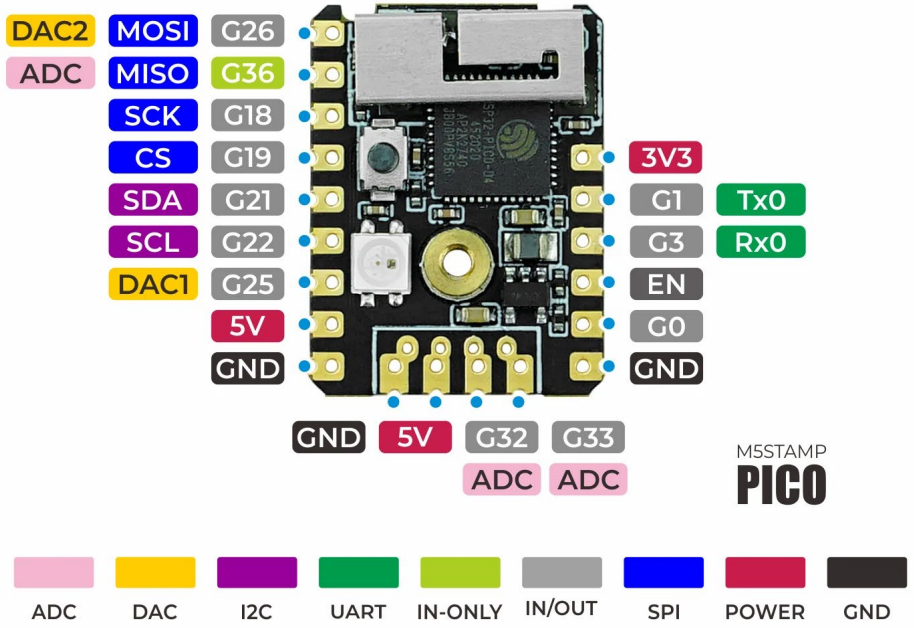
M5STACK



Neopixel	G27
Button	G39

M5STAMP
PICO

ADC DAC I2C UART IN-ONLY IN/OUT SPI POWER GND



M5Stack Stamp Pico	WeAct USB2UART
3V3	3V3
G1	RX
G3	TX
EN	RTS
G0	DTR
GND	GND

```
#define DS4 4
#define DS3 3
#define CONTROLLER DS4                      // quale joypad voglio usare: DualShock 3 o 4

#include "Cdrv8833.h"
#include "ESP32Servo.h"
#if CONTROLLER == DS4
#include "PS4Controller.h"
#elif CONTROLLER == DS3
#include "Ps3Controller.h"
#endif

// configurazione motore DC
#define MOTOR_PIN_A                 18      // pin al quale è connesso il pin IN1/IN3 del DRV8833
#define MOTOR_PIN_B                 19      // pin al quale è connesso il pin IN2/IN4 del DRV8833
#define MOTOR_CHANNEL                1      // canale utilizzato per la gestione del PWM -
                                            // DEVE ESSERE DIVERSO PER OGNI MOTORE
#define MOTOR_ROTATION_INVERTED false       // se vero, inverte il senso di rotazione del motore.
                                            // Utile se si è invertita la cablatura del motore/se il
              // motore è in posizione inversa rispetto all'altro
       // (coppia motore sinistro/destro)

#define MY_BT_ADDRESS  "00:11:22:33:44:55" // indirizzo bluetooth utilizzato per l'accoppiamento
                                            // tra l'ESP32 ed il joypad.
   // Inserisci il tuo valore definendo sei byte in formato
   // esadecimale, separati da due punti
   // Questo indirizzo deve essere uguale a quello settato
   // sul joypad mediante l'utility sixaxispairtool

// oggetto per la gestione del motore DC
Cdrv8833 DCMotor(MOTOR_PIN_A, MOTOR_PIN_B, MOTOR_CHANNEL, MOTOR_ROTATION_INVERTED);
```

```
// callback che verrà chiamata ogni qualvolta che il joypad si connette
void onConnect(void) {
   Serial.println("onConnect");
}

// callback che verrà chiamata goni qualvolta che il joypad si disconnette
void onDisconnect(void) {
   Serial.println("onDisconnect");
}

// callback che verrà chiamata ogni qualvolta che ci sarà un nuovo pacchetto dati inviato dal joypad
void onPacket(void) {

   int8_t ly, value;

   // leggo i valori dello stick sinistro del joypad
#if CONTROLLER == DS4
   ly = PS4.data.analog.stick.ly;
#elif CONTROLLER == DS3
   ly = Ps3.data.analog.stick.ly;
#endif

   // i valori degli stick variano tra -128 e +127;
   // la velocità di rotazione del motore varia tra
   // -100 (100% indietro) e 100 (100% avanti)
   value = map(ly, -128, 127, -100, 100);

   // aggiorno la velocità di rotazione del motore
   DCMotor.move(value);
}
```

```
void setup() {
   // inizializza la seriale e stampa un messaggio
   Serial.begin(115200);
   delay(1000);
   Serial.printf("\n\n---| ESP32 BattleBots controller |---\n\n");

   // setto la modalità slow decay per i motori DC. Questo permette di avere una coppia maggiore
   // sull'asse del motore rispetto a quanto potremmo ottenere con la modalità fast decay
   DCMotor.setDecayMode(drv8833DecaySlow);

   // inizializza la libreria per la gestione della comunicazione tra il joypad DS4 e la M5Stack Stamp Pico
#if CONTROLLER == DS4
   if (!PS4.begin(MY_BT_ADDRESS)) {
#elif CONTROLLER == DS3
   if (!Ps3.begin(MY_BT_ADDRESS)) {
#endif
      Serial.println("PS Library NOT initialized.");
      while (1);
   }
   else
   Serial.println("PS Library initialization done.");



// CONTINUA ----->
```

```
// <----- CONTINUA



// setto le callback che verranno chiamate ogni qualvolta che il joypad
// si connette/disconnette/ci sono nuovi pacchetti contenenti lo stato del joypad
// (pulsanti premuti, posizione stick, triggers, etc)
#if CONTROLLER == DS4
   PS4.attachOnConnect(onConnect);
   PS4.attachOnDisconnect(onDisconnect);
   PS4.attach(onPacket);
#elif CONTROLLER == DS3
   Ps3.attachOnConnect(onConnect);
   Ps3.attachOnDisconnect(onDisconnect);
   Ps3.attach(onPacket);
#endif
}

void loop() {
}
```

Se il controller si disconnette dall'ESP32 subito dopo essersi connesso

1) Scaricare esptool da qui:

   https://github.com/espressif/esptool/releases/latest

   Assicurarsi di scaricare la versione compatibile con il proprio Sistema Operativo

2) Decomprimere il pacchetto scaricato al punto 1)

3) Aprire un prompt dei comandi e spostarsi nella directory dove si è decompresso il pacchetto esptool

4) Assicurarsi che l'ESP32 sia connesso al PC con un cavo dati e che non ci siano monitor seriali aperti

5) Eseguire il comando

   ```
   esptool --chip esp32 erase_flash
   ```

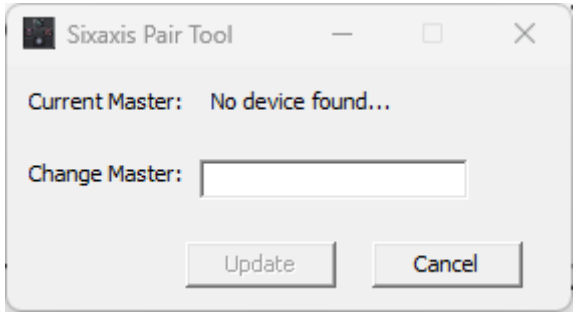
6) Ricaricare il firmware e riprovare a connettere il controller

Leggere/modificare l’indirizzo Bluetooth al quale il controller dovrà collegarsi

1) Installare SixaxisPairTool ed eseguirlo 

2) Connettere il controller al PC con un cavo dati USB

3) Modificare l’indirizzo Bluetooth inserendo il nuovo nel formato XX:XX:XX:XX:XX:XX nel campo Change Master e premere Update



OPPURE

Copiare l’indirizzo presente nel campo Current Master ed inserirlo all’interno dello sketch nella funzione membro

```
Ps3.begin(<indirizzo bt>)
```

OPPURE

```
PS4.begin(<indirizzo bt>)
```