IOT PROTO SHIELD PLUS USER MANUAL

# IOT PROTO SHIELD PLUS (VER. 1.0) USER MANUAL



#### INDEX

1.	Introduction and technical specifications	7
2.	Getting the IDT PROTO SHIELD PLUS board out of the box	9
3.	Layout	11
4.	Important considerations about 5V and 3V3 before starting to work with the IDT PROTO SHIEL	_D
PLUS	s board	14
4.1.	Switching Vbrd to 5V	14
5.	Using the IDT PROTO SHIELD PLUS with Arduino NANO and NANO EVERY boards	16
6.	Using the IDT PROTO SHIELD PLUS with ESP32 Mini boards	17
7.	Using the IDT PROTO SHIELD PLUS with ESP8266 Mini boards	19
8.	Default connections on the IDT PROTO SHIELD PLUS	20
9.	Using the IoT Proto Shield Plus with other form factor boards	22
9.1.	UNO to IoT Proto Shield Plus Adapter	22
9.2.	ESP32 or ESP8266 boards with different form factor	23
10.	LEDs	24
10.1.	Default connections	25
10.2.	Sample Sketches	25
10.3.	Using the LEDs with different GPIO pins	25
11.	POT	26
11.1.	Default connections	27
11.2.	Sample Sketches	27
11.3.	Using the POT with a different GPIO pin	27
12.	ENCODER	28
12.1.	Default connections	29
12.2.	Suggested ESP8266 connections	29
12.3.	Sample Sketches	30
12.4.	Using the ENCODER with different GPIO pins	30
13.	ТОՍСН	31
13.1.	Default connections	32
13.2.	Suggested ESP8266 connections	32
13.3.	Sample Sketches	32
13.4.	Using TOUCH with a different GPIO pin	32

14.	I <sup>2</sup> C on the IoT Proto Shield Plus	. 34
14.1.	Default connections	. 34
14.2.	Scanning for I2C devices	. 34
14.3.	Using the I <sup>2</sup> C with different GPIO pins	. 34
15.	SSD1306 128x64 OLED DISPLAY (I <sup>2</sup> C type)	. 36
15.1.	Default connections	. 37
15.2.	Sample Sketches	. 37
15.3.	IoTPSP_SSD1306_DISPLAY_128x64_NANO	. 37
15.4.	IoTPSP_SSD1306_DISPLAY_128x64_ESP32	. 37
15.5.	IoTPSP_SSD1306_DISPLAY_128x64_ESP8266	. 38
15.6.	Using the SSD1306 128x64 OLED display with different GPIO pins	. 38
16.	SSD1306 128x32 OLED DISPLAY (I <sup>2</sup> C type)	. 39
16.1.	Default connections	. 40
16.2.	Sample Sketches	. 40
16.3.	IoTPSP_SSD1306_DISPLAY_128x32_NANO	. 40
16.4.	IoTPSP_SSD1306_DISPLAY_128x32_ESP32	. 40
16.5.	IoTPSP_SSD1306_DISPLAY_128x32_ESP8266	. 40
16.6.	Using the SSD1306 128x32 OLED display with different GPIO pins	. 40
17.	LCD 1602 (Liquid Crystal Display)	. 41
17.1.	Default connections	. 42
17.2.	Sample Sketches	. 42
17.3.	Changing the I <sup>2</sup> C address of the PCF8574	. 43
17.4.	Using the PCF8574 with different I <sup>2</sup> C GPIO pins	. 43
18.	PCF8574 I2C I/O expander	. 43
18.1.	Sample Sketches	. 44
18.2.	IoTPSP_PCF8574_GET_ADDRESS	. 44
19.	TM1637 Digital display module	. 46
19.1.	Default connections	. 47
19.2.	Sample Sketches	. 47
19.3.	Using the TM1637 display with different GPIO pins	. 48
20.	RELAY	. 49
20.1.	Default connections	. 50
20.2.	Suggested ESP8266 connections	. 50

21. 21.7. Using DHT with a different GPIO pin......55 22. 23. 24 25. Serial communication on the IDT PROTO SHIELD PLUS 26 27. 27.3. IoTPSP\_MP3\_NANO ...... 68 27.4. IoTPSP\_MP3\_NANO\_EVERY\_33......68

27.5.	IoTPSP_MP3_ESP32	. 68
27.6.	IoTPSP_MP3_ESP8266	. 69
27.7.	Using the DFPlayer Mini MP3 module with a different GPIO pins	. 69
27.8.	IoTPSP_MP3_NANO_SoftwareSerial	. 69
28.	SPEAKER	. 70
28.1.	Default connections	. 71
28.2.	Sample Sketches	. 71
28.3.	IoTPSP_SPEAKER_NANO	. 71
28.4.	IoTPSP_SPEAKER_ESP32	. 71
28.5.	IoTPSP_SPEAKER_ESP8266	. 72
29.	MIC	. 72
29.1.	MAX9814 microphone module	. 73
29.2.	Setting the Gain	. 74
29.3.	Setting the module accuracy	. 74
29.4.	MAX4466 microphone module	. 74
29.5.	Default connections	. 75
29.6.	Suggested ESP8266 connections	. 75
29.7.	Sample Sketches	. 75
29.8.	Using the MIC with a different GPIO pin	. 75
30.	SERVO	. 77
30.1.	Default connections	. 77
30.2.	Suggested ESP8266 connections	. 78
30.3.	Sample Sketches	. 78
30.4.	IoTPSP_SERVO_NANO	. 78
30.5.	IoTPSP_SERVO_ESP32	. 79
30.6.	IoTPSP_SERVO_ESP8266	. 80
30.7.	Using SERVO with a different GPIO pin	. 80
30.8.	Powering the SERVO with a different power supply	. 80
31.	HC-SR04 (distance sensor)	. 81
31.1.	Default connections	. 82
31.2.	Suggested ESP8266 connections	. 83
31.3.	Sample Sketches	. 83
31.4.	Using HC-SR04 with different GPIO pins	. 83

IoT
PROTO SHIELD PLUS
USER MANUAL

32.	STEPPER MOTOR	84
32.1.	Default connections	85
32.2.	Suggested ESP8266 connections	86
32.3.	Sample Sketches	86
32.4.	IoTPSP_STEPPER_MOTOR_NANO	86
32.5.	IoTPSP_STEPPER_MOTOR_ESP32	87
32.6.	IoTPSP_STEPPER_MOTOR_ESP8266	87
32.7.	Choosing the Stepper Driver Module	88
32.8.	DRV8825 module	88
32.9.	TMC2xxx module	89
32.10	.DRV8834 module	90
32.11	.A4988 module	91
32.12	.Choosing and connecting the motor	93
32.13	.Using STEPPER MOTOR with different GPIO pins or use additional features	93
33.	LEVEL SHIFTER	94
33.1.	Using the onboard level shifter with different HVin and / or LVin	95
34.	ESP01 WiFi module	97
35.	EXTERNAL POWER SUPPLY	99
35.1.	Battery Shield 1	.01
36.	The IoT Proto Shield Plus schematic design 1	03

#### NOTE:

Some pictures contained in this manual, may be different from the board revision in your hands (at time of writing the current version is rev. 1.0), this does not affect the functionality of the board described in this manual.

# 1. Introduction and technical specifications

The **IOT PROTO SHIELD PLUB** is designed to let you easily prototype with ESP32, ESP8266 and ARDUINO NANO boards.

It works with:

- Arduino NANO form factor boards
  - o Arduino NANO 33 IoT
  - o Arduino NANO 33 BLE
  - o Arduino Nano RP2040 Connect
  - o Arduino NANO
  - o <u>Arduino NANO EVERY</u>
- ESP32 MINI form factor boards
- ESP8266 MINI form factor boards

#### Features:

.

- Wide breadboarding area
  - 2x solderless breadboarding areas 740 contact points each (total 1480 points)
  - Breakout headers to replicate each I/O pin close to the breadboarding area
- Breakout headers to replicate 3V3, 5V and GND close to the breadboarding area
- Different working voltages:
  - o 3V3
  - o 5V
  - external power supply by means of an additional LM2596 DC-DC module of widespread use
  - external power supply by means of an additional Battery Shield of widespread use
- Onboard, ready to use, rotary encoder with switch
- Onboard, ready to use, touch switch
- Two onboard, ready to use, OUTPUT LEDs
- Onboard, ready to use, potentiometer with knob
- Onboard speaker to play MP3 or tones
- Onboard, ready to use, PCF8574 I/O expander to drive 16x2 LCD (I2C communication with LCD with just 2 pins)
- Free soldering pads for additional prototyping parts
- Four onboard, ready to use, level shifter
- RST button, for resetting program
- Plug and play predisposition for the following modules, sensors and actuators of widespread use:
  - Display LCD 16x2
  - o Display OLED SSD1306 128x64
  - Display OLED SSD1306 128x32
  - TM1637 seven-segment display
  - o RTC (Real Time Clock)
  - o SD Card
  - o Servo
  - LDR (Light Dependant Resistor)
  - PTC or NTC (Positive or Negative Temperature Coefficient thermistor)
  - o RELAY
  - HC-SR04 (ultrasonic distance sensor)
  - DTH11 / DHT22 module or DHT11 / DHT22 bare sensor (humidity sensor)

- MAX9814 or MAX4466 module (microphone module)
- DFR0299 MP3 Mini Player (MP3 file music player)
- Board dimensions in mm: 188x196x30<sup>1</sup> (HxWxH)

<sup>&</sup>lt;sup>1</sup> 188x196x30 without any other additional board plugged into

# Getting the IoT PROTO SHIELD PLUS board out of the box

The IOT PROTO SHIELD PLUS consists of:

- 1x IoT PROTO SHIELD PLUS board
- 5x adhesive rubber feet

2.

• 1x Potentiometer knob

All you need to do is just to insert the knob in the potentiometer (Figure 1) and stick the adhesive feet (Figure 2) in case that they are not already sticked (it depends on the production process flow).



Figure 1 – Placing the potentiometer knob



Figure 2 – Sticking the five adhesive rubber feet

IoT Proto Shield Plus User Manual

# 3. Layout



Figure 3 – Layout of the IOT PROTO SHIELD PLUS

Please note that the **IDT PROTO SHIELD PLUB** works with just one Controller Board at time (i.e. you can't plug the ESP32 and the NANO at the same time).

#### IOT PROTO SHIELD PLUS USER MANUAL



Figure 4 – Top view of the IOT PROTO SHIELD PLUS



Figure 5 – Bottom view of the IOT PROTO SHIELD PLUS

# 4. Important considerations about 5V and 3V3 before starting to work with the IOT PROTO SHIELD PLUS board

ESP32, ESP8266, NANO and NANO33 boards have an operating voltage of 5V or 3,3V (from now on 3V3). A 5V operating voltage device can handle a 3V3 input signal on its pins, while a 3V3 operating voltage device may be damaged if connected to a 5V input signal.

That's why the **IDT PROTO SHIELD PLUS** routes 3V3 voltage as the default voltage to power all the devices that you can plug in the dedicated headers.

Vbrd sets and routes the operating voltage of the IoT PROTO SHIELD PLUS.



Figure 6 – Schematic of the routing of 3V3 or 5V to Vbrd

## 4.1. Switching Vbrd to 5V

Be very careful if you switch Vbrd to 5V because you can damage some 3V3 operating voltage devices.

To switch Vbrd to 5V: OPEN (cut in the middle) the connection between JP1 pad 1 and pad 2 and CLOSE (connect with a little drop of tin) JP1 pad 2 to pad 3 (see Figure 9).



Figure 7 – JP1 on the IoT PROTO SHIELD PLUS



Figure 8 – Setting Vbrd to 3V3 (default)



Figure 9 – Setting Vbrd to 5V

# 5. Using the IOT PROTO SHIELD PLUS with Arduino NANO and NANO EVERY boards

Plug the NANO board in the Arduino NANO SOCKET paying attention to pin alignment and orientation. Connect the NANO board by means of a USB cable to your PC and start using it. Consider that NANO and NANO EVERY boards operate at 5V, while NANO33 family boards operate at 3V3.

It is strongly suggested to read the technical specification of the board you are using to become familiar with it.



Figure 10 - An ARDUINO NANO (NANO33 IoT) plugged into the IoT PROTO SHIELD PLUS



Figure 11 – ARDUINO NANO signals routing on the IDT PROTO SHIELD PLUS

# 6. Using the IOT PROTO SHIELD PLUS with ESP32 Mini boards

Plug the ESP32 MINI board in the ESP SOCKET paying attention to pin alignment and orientation. Connect the ESP32 MINI board by means of a USB cable to your PC and start using it. Consider that ESP32 boards operate at 3V3.

It is strongly suggested to read the technical specification of the board you are using to become familiar with it.





Figure 12 – The ESP32 mini D1 plugged into the IoT PROTO SHIELD PLUS



Figure 13 – ESP32 signals routing on the IDT PROTO SHIELD PLUS

# 7. Using the IoT PROTO SHIELD PLUS with ESP8266 Mini boards

Consider that the footprint of the ESP8266 mini board is a "shrinked" version of the ESP32 footprint (or the ESP32 is an "extended" version of the ESP8266 footprint....) and they are compatible since they share the pinout on some pins.

Plug the ESP8266 MINI board in the ESP SOCKET paying attention to pin alignment and orientation. Connect the ESP8266 MINI board by means of a USB cable to your PC and start using it. Consider that ESP8266 boards operate at 3V3.

It is strongly suggested to read the technical specification of the board you are using to become familiar with it.





Figure 14 - The ESP8266 D1 mini plugged into the IOT PROTO SHIELD PLUS

# 8. Default connections on the IDT PROTO SHIELD PLUS

В	BOARDS PINOUT			ROTO SHIELI	D PLUS
ESP 8266	ESP32 Mini	Arduino NANO	Primary Function	Secondary Function	Reference JP
	GPIO00 / 100				
GPIO1 / TX	GPIO01 / TXD				
	GPIO02 / IO2				
GPIO3 / RX	GPIO03 / RXD				
m	GPIO04 / IO4	D5	Servo		JP53
GPIO15 / D8	GPIO05 / 105	D10	SPI_CS		
	GPIO12 / TDI	D8	RELAY	JTAG_TDI	JP51
	GPIO13 / TCK	D9	DHT	JTAG_TCK	JP56
	GPIO14 / TMS	D6	HC_SR_T	JTAG_TMS	JP39
	GPIO15 / TD0	D7	HC_SR_E	JTAG_TD0	JP40
GPIO2 / D4	GPIO16 / IO16	D0	RX		JP46
GPIO0 / D3	GPIO17 / IO17	D1	тх	SPEAKER	JP47
GPIO14 / D5	GPIO18 / IO18	D13	SPI_SCLK		
GPIO12 / D6	GPIO19 / IO19	D12	LED-GRN	SPI_MISO	JP11
GPIO4 / D2	GPIO21 / 1021	A4	I2C_SDA		JP7
GPIO5 / D1	GPIO22 / 1022	A5	I2C_SCL		JP6
GPIO13 / D7	GPIO23 / IO23	D11	LED_RED	SPI_MOSI	JP9
	GPIO25 / 1O25	D4	MOT_DIR		JP37
GPIO16 / D0	GPIO26 / IO26	D3	MOT_STP		JP38
	GPIO27 / 1027	D2	ENC-A		JP20
	GPIO32 / 1032	A0	ТОИСН		JP3
	GPI033 / 1033	A1	ENC-SW		JP18
	GPIO34 / 1034	A2	LDR / xTC		JP50
	GPIO35 / 1035	A3	ENC-B		JP19
ADC0	GPIO36 / SVP	A6	РОТ		JP12
	GPIO39 / SVN	A7	MIC		JP28

#### HIGH ON BOOT

BOOT FAILURE IF PULLED LOW

BOOT FAILURE IF PULLED HIGH

CONNECTED TO ONBOARD DEVICES BY DEFAULT

FREE TO BE USED BY DEFAULT

#### HIGH ON BOOT

You have to consider that during the device booting process, the pin will be in high state.

i.e.

If you are using an ESP8266 module and you want to use its pin GPIO16/D0, you have to consider that the pin GPIO16/D0 will be in high state during the booting process of the ESP8266 module.

#### BOOT FAILURE IF PULLED LOW

You have to consider that, if you connect the pin to a PULL DOWN resistor, the device will not boot (it will not work).

i.e.

If you are using an ESP8266 module and you connect pin GPIO0/D3 to a pull down resistor, the ESP8266 module will not boot, thus it will not work.

#### BOOT FAILURE IF PULLED HIGH

You have to consider that, if you connect the pin to a PULL UP resistor the device will not boot (it will not work).

i.e.

If you are using an ESP32 module and you connect pin GPIO12/TDI to a pull up resistor, the ESP32 module will not boot, thus it will not work.

#### CONNECTED TO ONBOARD DEVICES BY DEFAULT

You have to consider that the pin is connected by default to one of the onboard devices of the IDT **PROTO SHIELD PLUS**.

i.e.

If you are using an Arduino NANO and you want to use A6 to acquire an analogue signal, you have to consider that it connects by default to the onboard POT (RV2).

To use A6 you need to disconnect it from POT cutting JP12 on the middle (see further in this manual detailed description for each onboard device).

#### FREE TO BE USED BY DEFAULT

If its primary function device or module is not inserted in the **IDT PROTO SHIELD PLUS**, you can use the pin for your own purposes without any trouble.

i.e.

If you are using an Arduino NANO and you want to use A7 to acquire an analogue signal, and the MIC module is not plugged into the MIC header, you can connect your signal to A7 without any trouble.

#### CONCLUSION

- If you need to use one device or module, and the **IDT PROTO SHIELD PLUS** has the plug and play predisposition for that module, just plug it into the header (see further in this manual the detailed description for each plug and play predisposition) and it will be ready to work.
- If you need to use a device or module, and the IDT PROTO SHIELD PLUS does not have the plug and play predisposition for that module, plug it into the breadboarding area and connect it to one of the "FREE TO BE USED BY DEFAULT" pin. Remember to consider its possible behavior during boot.
- If you absolutely need to use one of the "CONNECTED TO ONBOARD DEVICES BY DEFAULT" pin, you must disconnect it from the onboard device before using it for other purposes. Remember to consider its possible behavior during boot.

If you need to use the **IDT PROTO SHIELD PLUS** with a board with a form factor different than NANO (see §5) or the ESP32 D1 MINI (see §6) or the ESP8266 D1 MINI (see §7), you need to use an adapter.

# 9.1. UNO to IoT Proto Shield Plus Adapter

If you need to use the IoT PROTO SHIELD PLUS with an Arduino UNO form factor board (i.e. Arduino UNO, Leonardo, Zero, Yun, etc.), you can use the UNO to IoT Proto Shield Plus Adapter (https://www.gtronicsshop.com/en/proto-shield-plus/37-uno-to-iot-proto-shield-plus-adapter.html).

The adapter works with any board compatible with Arduino UNO in terms of form factor, pin-out and electrical specifications.

Anyway, before using the Uno To IoT Proto Shield Plus Adapter, please take a look at the user manual.



Figure 15 – The UNO to IoT Proto Shield Plus Adapter

# 9.2. ESP32 or ESP8266 boards with different form factor

If you need to use the **IDT PROTO SHIELD PLUS** with an ESP32 or ESP8266 board with a different form factor (i.e. the ones shown in Figure 16) you can request us if an adapter is available or design your own adapter taking as an example the UNO to IOT Proto Shield Plus adapter shown in the previous paragraph.



Figure 16 – Examples of ESP32 and ESP8266 boards

IOT PROTO SHIELD PLUS USER MANUAL

# 10. LEDs

Working with the two onboard LEDs of the **IOT PROTO SHIELD PLUS**.





Figure 17 – LEDs Section of the IDT PROTO SHIELD PLUS



Figure 18 – Schematic of the IDT PROTO SHIELD PLUS LEDs Section

#### **10.1.** Default connections

	ESP32	ARDUINO	ESP8266		
GRN LED	GPIO19	D12	GPIO12 (D6)		
RED LED	GIOP23	D11	GPIO13 (D7)		

## **10.2.** Sample Sketches

Location: IoTPSP\_Sample\_Sketches\IoTPSP\_LEDs

According to the board you are using and the signals routing, the sketches make the green (GRN) and the red (RED) LEDs blink.

Use the IoTPSP\_LEDs\_NANO sketch to test the LEDs with ARDUINO NANO boards. Use the IoTPSP\_LEDs\_ESP32 sketch to test the LEDs with ESP32 boards. Use the IoTPSP\_LEDs\_ESP8266 sketch to test the LEDs with ESP8266 boards.

## 10.3. Using the LEDs with different GPIO pins

According to Figure 18, GRN and RED LEDs are routed to the default GPIO pins by means of JP11 and JP9. To disconnect GRN LED from its default GPIO, you must open JP11 (cut it in the middle).

To disconnect RED LED from its default GPIO, you must open JP9 (cut it in the middle).

Solder a header on J6 and, by means of some jumper wires, connect GRN and RED LEDs to other GPIO pins according to your needs.

Please check the default signal routing of the **IDT PROTO SHIELD PLUS** (see §8) to avoid conflict with other default connections.

IOT PROTO SHIELD PLUS USER MANUAL

# 11. POT

Working with the onboard Potentiometer of the IDT PROTO SHIELD PLUS.





Figure 19 – POT section of the IoT PROTO SHIELD PLUS





#### 11.1. Default connections

	ESP32	ARDUINO	ESP8266
РОТ	GPIO36	A6	A0

## 11.2. Sample Sketches

Location: IoTPSP\_Sample\_Sketches\IoTPSP\_POT

According to the board you are using and the signals routing, the sketches read the analog value of the POT and print it to the serial monitor.

Use the IoTPSP\_POT\_NANO sketch to test the POT with ARDUINO NANO boards. Use the IoTPSP\_POT\_ESP32 sketch to test the POT with ESP32 boards. Use the IoTPSP\_POT\_ESP3266 sketch to test the POT with ESP8266 boards.

# 11.3. Using the POT with a different GPIO pin

According to Figure 20, POT is routed to the default GPIO pin by means of JP12.

To disconnect POT from its default GPIO pin, you must open JP12 (cut it in the middle).

Solder a header on J7 and, by means of some jumper wires, connect it to another GPIO pin according to your needs.

Please check the default signal routing of the **IDT PROTO SHIELD PLUS** (see §8) to avoid conflict with other default connections.

IOT PROTO SHIELD PLUS USER MANUAL

# **12. ENCODER**

Working with the onboard ENCODER of the IDT PROTO SHIELD PLUS.





Figure 21 - ENCODER section of the IDT PROTO SHIELD PLUS



Figure 22 – Schematic of the IOT PROTO SHIELD PLUS ENCODER section

#### 12.1. Default connections

	ESP32	ESP8266		
ENC A	GPIO27	D2	NC	
ENC B	GPIO35	A3	NC	
ENC SW	GPIO33	A1	NC	

*NC* = Not Connected

Please note that there are no default connections for the ESP8266. If you need to use the ENCODER with ESP8266 boards, you need to route the signals with jumper wires.

## 12.2. Suggested ESP8266 connections

The sample sketches are made to work with these connections: ESP8622 GPIO 14 (D5) connected to header J3 ESP32 GPIO33 (ENC\_SW) ESP8622 GPIO 5 (D1) connected to header J3 ESP32 GPIO27 (ENC\_A) ESP8622 GPIO 4 (D2) connected to header J3 ESP32 GPIO35 (ENC\_B)



## 12.3. Sample Sketches

Location: IoTPSP\_Sample\_Sketches\IoTPSP\_ENCODER

Use the IoTPSP\_ENCODER\_Simple\_NANO sketch to test the ENCODER with ARDUINO NANO boards. Use the IoTPSP\_ENCODER\_Simple\_ESP32 sketch to test the ENCODER with ESP32 boards. Use the IoTPSP\_ENCODER\_Simple\_ESP8266 sketch to test the ENCODER with ESP8266 boards.

Open the serial monitor and rotate the encoder knob to see its incremental position changing or press the encoder knob to see the GRN led changing its status.

A more robust way to manage the A and B signals of the ENCODER is using the interrupts.

Use the IoTPSP\_ENCODER\_Interrupt\_NANO sketch, to test the ENCODER with ARDUINO NANO boards using interrupts.

Use the IoTPSP\_ENCODER\_Interrupt\_ESP32 sketch, to test the ENCODER with ESP32 boards using interrupts.

Use the IoTPSP\_ENCODER\_Interrupt\_ESP8266 sketch, to test the ENCODER with ESP8266 boards using interrupts.

#### 12.4. Using the ENCODER with different GPIO pins

According to Figure 22, ENCODER signals are routed to the default GPIO pins by means of JP18, JP19 and JP20.

To disconnect ENCODER signals from their default GPIO pins, you must open JP 18, JP19 and JP20 (cut them in the middle).

Solder a header on J9 and, by means of some jumper wires, connect other GPIO pins according to your needs.

Please check the default signal routing of the **IDT PROTO SHIELD PLUS** (see §8) to avoid conflict with other default connections.

# 13. TOUCH

The touch sensor on the **IDT PROTO SHIELD PLUB** uses the TTP223E chip, which is the same used on the touch boards of widespread use.





Figure 24 – TOUCH section of the IDT PROTO SHIELD PLUS



Figure 25 – Schematic of the IoT PROTO SHIELD PLUS TOUCH section

Rev.1.07

#### 13.1. Default connections

	ESP32	ARDUINO	ESP8266					
TOUCH	GPIO32	A0	NC					

*NC* = Not Connected

Please note that there are no default connections for the ESP8266. If you need to use the TOUCH with ESP8266 boards, you need to route the signals with jumper wires.

#### **13.2.** Suggested ESP8266 connections

The sample sketch is made to work with this connection: ESP8622 GPIO 14 (D5) connected to header J3 ESP32 GPIO32 (TOUCH)



Figure 26 – Suggested connections to work with TOUCH and ESP8266

## **13.3.** Sample Sketches

Location: IoTPSP Sample Sketches\IoTPSP TOUCH

Use the IoTPSP\_TOUCH\_NANO sketch to test the TOUCH section with ARDUINO NANO boards. Use the IoTPSP\_TOUCH\_ESP32\_EXTERNAL sketch to test the TOUCH section with ESP32 boards. Use the IoTPSP\_TOUCH\_ESP8266 sketch to test the TOUCH section with ESP8266 boards.

Put your finger on the touch circles and the GRN led will turn on according to your touch.

ESP32 also have internal touch sensors, so with ESP32 boards, you can bypass the TTP223E chip and work directly with the internal sensor connected to GPIO32 (TOUCH9).

To setup the board to work with ESP32 internal touch sensor, you need to OPEN JP22 (cut it in the middle), CLOSE JP25 pad 3 and pad 2 (connect them with a little drop of tin) and OPEN pad 1 and 2 (cut between the pads).

Use the IoTPSP\_TOUCH\_ESP32\_INTERNAL sketch to test the TOUCH section with ESP32 boards. Use serial monitor to check the internal touch sensor value.

## 13.4. Using TOUCH with a different GPIO pin

According to Figure 25, TOUCH is routed to the default GPIO pin by means of JP3.

To disconnect TOUCH from its default GPIO pin, you must OPEN JP3 (cut it in the middle).

Solder a header on J43 and, by means of some jumper wires, connect it to another GPIO pin according to your needs.

Please check the default signal routing of the **IDT PROTO SHIELD PLUS** (see §8) to avoid conflict with other default connections.

# 14. I<sup>2</sup>C on the IoT Proto Shield Plus

I<sup>2</sup>C (Inter-Integrated Circuit, eye-squared-C) is widely used for attaching peripheral ICs to processors and microcontrollers (check out more about I<sup>2</sup>C <u>Wikipedia</u> or other online documentation).

I<sup>2</sup>C on the IOT PROTO SHIELD PLUS is used to communicate with devices such as displays, RTC clock, etc.



Figure 27 – I<sup>2</sup>C signals routing on the IoT PROTO SHIELD PLUS

## 14.1. Default connections

I<sup>2</sup>C signals on the IDT PROTO SHIELD PLUS are routed by default according to the following connections:

	ESP32	ARDUINO	ESP8266
I2C_SCL	GPIO22	A5	GPIO5 (D2)
I2C_SDA	GPIO21	A4	GPIO4 (D1)

# 14.2. Scanning for I2C devices

You can use the sample sketch IoTPSP\_I2C\_scanner (location:

IoTPSP Sample Sketches\IoTPSP I2C scanner) to scan I<sup>2</sup>C.

The scanner shows connected I<sup>2</sup>C devices addresses. With no external devices (such as OLED displays etc.) the scanner shows the address of the PCF8574 onboard chip (usually 0x20 or 0x38).

# **14.3.** Using the I<sup>2</sup>C with different GPIO pins

Most of the libraries used to communicate to  $I^2C$  devices, use the default  $I^2C$  GPIO pins (see §14.1), thus **it is NOT recommended to reroute I2C signals**.

Anyway, if you want or need, follow the instructions below.

According to Figure 27, I<sup>2</sup>C signals are routed to the default GPIO pins by means of JP6 and JP7.

To disconnect I<sup>2</sup>C signals from their default GPIO pins, you must OPEN JP6 and JP7 pads 1-2 (cut in the middle of pad 1 and pad 2) and CLOSE pads 2 with pad 3 (by means of a little drop of tin). Solder a header on J41 and, by means of some jumper wires, connect other GPIO pins according to your needs.

Please check the default signal routing of the **IDT PROTO SHIELD PLUS** (see §8) to avoid conflict with other default connections.

# 15. SSD1306 128x64 OLED DISPLAY (I<sup>2</sup>C type)

The IOT PROTO SHIELD PLUS is ready to work with 4 pins I<sup>2</sup>C SSD1306 128x64 OLED displays of widespread use.

To work with a I<sup>2</sup>C SSD1306 128x64 OLED display, just plug the module into the BRD9 header according to Figure 30.



Figure 28 – BRD9 OLED DISPLAY 128x64 header of the IOT PROTO SHIELD PLUS



Figure 29 – Schematic of the IOT PROTO SHIELD PLUS BRD9 OLED DISPLAY 128x64 header

LCD 16x	v55 VD6	2 2 2	3 <sup></sup> 8	5 5	8 8	20	<u>۱</u> ۳۱۹ <u>۱</u> ۱۹			9.4 SDA	Ő
00000				00000		00000	SDA A SCK Vcc C GND B-w10	1 3 2 3			
	000			000	000	0000		0	J		0

Figure 30 – An I<sup>2</sup>C SSD1306 128x64 OLED display module plugged into BRD9 header of the IDT PROTO SHIELD PLUS
#### 15.1. Default connections

Refer to § 14.1 to see default routing of I<sup>2</sup>C signals on the IOT PROTO SHIELD PLUS.

#### 15.2. Sample Sketches

Location: IoTPSP\_Sample\_Sketches\IoTPSP\_SSD1306\_DISPLAY\_128x64

#### 15.3. IoTPSP\_SSD1306\_DISPLAY\_128x64\_NANO

The IoTPSP\_SSD1306\_DISPLAY\_128x64\_NANO sketch, used to test the IoT PROTO SHIELD PLUS with NANO boards, uses the *Adafruit SSD1306* library. Be sure to install it before using the sketch.

	Auto Format	Ctrl+T	
	Archive Sketch		
IoTPSP_SSD1	Fix Encoding & Reload		
******	Manage Libraries	Ctrl+Shift+I	
this is an the sketch m	Serial Monitor	Ctrl+Shift+M	
Library Manager	100000		
11000			
ACROBOTIC SSD134 by ACROBOTIC SSD134 by ACROBOTIC vo Library for SSD134 More info	Topic I All      1305     1305     130	SSD1306-powered OLED 128x64 displays	
AL ACROBOTIC SSD130 Sy ACROBOTIC Vo Library for SSD131 More info	(Topic All 1306 Service 1.0.1 INSTALLED De-powered OLED 128x04 displayst Library for	SSD1308-powered OLED 128x64 displays	
ACROBOTIC SSD134 by ACROBOTIC SSD134 by ACROBOTIC Vo Liberar for SSD134 More info SSD1306 eiled driv and 128:32 displa More info SSD1306 eiled driv and 128:32 displa More info Select version v	Top: (All 2006) Top 2006 Top owered OLED 128x04 displayst Library for n 2.4-3 INSTALLED to 2.4-3 INSTALLED	SSD1308-powersd OLED 128x64 displays SSD1308-powersd OLED 128x64 displays	nenechram= 128=54 Ubdate

## 15.4. IoTPSP\_SSD1306\_DISPLAY\_128x64\_ESP32

The IoTPSP\_SSD1306\_DISPLAY\_128x64\_ESP32 sketch, used to test the IoT PROTO SHIELD PLUS with ESP32 boards, uses the *ESP8266 and ESP32 OLED driver for SSD1306 displays* library. Be sure to install it before using the sketch.

IoTPSP_SSD1	306_DISPLAY_128x64_NANO   Arduino 1.8.13	
File Edit Sketch	Tools Help	
	Auto Format	Ctrl+T
	Archive Sketch	
IoTPSP_SSD1	Fix Encoding & Reload	
/********	Manage Libraries	Ctrl+Shift+I
This is an the sketch m	Serial Monitor	Ctrl+Shift+M

S Library Manager	×
Type All V Topic All V 1306	
C3P Generate QNCOUE for GDCH0Z13572 emix uisplay, 5301300, 541100 oleu uisplays 120°04 pixel and others based on Adafruit 5772XX ESP Generate QRCode for GDEH0213872 eink display, SSD1306, SH1106 oled displays 128*64 pixel and others based on Adafruit ST77XX More info	*
ESP8266 and ESP32 OLED driver for SSD1306 displays	1
by ThingPulse, Fabrice Weinberg Version 4.2.0 INSTALLED 12C display driver for SSD1306 OLED displays connected to ESP8266, ESP32, Mbed-OS The following geometries are currently supported 128x64, 128x32, 64x48. The init sequence was inspired by Adafruit's library for the same display. More info Select version  Select version  Install	
by Anun Panya ESP8266 Generate QRCode for SSD1306 oled displays 128*64 pixel ESP8266 Generate QRCode version 7 for SSD1306 oled displays 128*64 pixel <u>More info</u>	~
Close	2

#### 15.5. IoTPSP\_SSD1306\_DISPLAY\_128x64\_ESP8266

The IoTPSP\_SSD1306\_DISPLAY\_128x64\_ESP8266 sketch, used to test the IoT PROTO SHIELD PLUS with ESP8266 boards, uses the *ESP8266 and ESP32 OLED driver for SSD1306 displays* library. Be sure to install it before using the sketch (see § 15.4).

#### 15.6. Using the SSD1306 128x64 OLED display with different GPIO pins

Refer to § 14.3 to see how to use I<sup>2</sup>C with different GPIO pins on the **IOT PROTO SHIELD PLUS.** 

## 16. SSD1306 128x32 OLED DISPLAY (I<sup>2</sup>C type)

The IOT PROTO SHIELD PLUS is ready to work with 4 pins I<sup>2</sup>C SSD1306 128x32 OLED displays of widespread use.

To work with a I<sup>2</sup>C SSD1306 128x32 OLED display, just plug the module into the BRD10 header according to Figure 33.



Figure 31 – BRD10 OLED DISPLAY 128x32 header of the IOT PROTO SHIELD PLUS



Figure 32 – Schematic of the IoT PROTO SHIELD PLUS BRD10 OLED DISPLAY 128x32 header

LCD 16k 성 응 위 입 중 <sup>11</sup> 입 중 11 명	2 5 5 5 5	JRD9 LCD
		. 0

Figure 33 – An I<sup>2</sup>C SSD1306 128x32 OLED display module plugged into BRD10 header of the IoT PROTO SHIELD PLUS

#### **16.1.** Default connections

Refer to § 14.1 to see default routing of I<sup>2</sup>C signals on the **IOT PROTO SHIELD PLUS.** 

#### 16.2. Sample Sketches

Location: IoTPSP Sample Sketches\IoTPSP SSD1306 DISPLAY 128x32

#### 16.3. IoTPSP\_SSD1306\_DISPLAY\_128x32\_NANO

The IoTPSP\_SSD1306\_DISPLAY\_128x32\_NANO sketch, used to test the IoT PROTO SHIELD PLUS with NANO boards, uses the *Adafruit SSD1306* library. Be sure to install it before using the sketch (see § 15.3).

#### 16.4. IoTPSP\_SSD1306\_DISPLAY\_128x32\_ESP32

The IoTPSP\_SSD1306\_DISPLAY\_128x32\_ESP32 sketch, used to test the IoT PROTO SHIELD PLUS with ESP32 boards, uses the *ESP8266 and ESP32 OLED driver for SSD1306 displays* library. Be sure to install it before using the sketch (see § 15.4).

#### 16.5. IoTPSP\_SSD1306\_DISPLAY\_128x32\_ESP8266

The IoTPSP\_SSD1306\_DISPLAY\_128x32\_ESP8266 sketch, used to test the IoT PROTO SHIELD PLUS with ESP8266 boards, uses the *ESP8266 and ESP32 OLED driver for SSD1306 displays* library. Be sure to install it before using the sketch (see § 15.4).

#### 16.6. Using the SSD1306 128x32 OLED display with different GPIO pins

Refer to § 14.3 to see how to use I<sup>2</sup>C with different GPIO pins on the IOT PROTO SHIELD PLUS.

## 17. LCD 1602 (Liquid Crystal Display)

The **IDT PROTO SHIELD PLUS** is ready to work with 16 pins LCD 1602 displays of widespread use. To work with an LCD 1602 display, just plug the module into the LCD16x2 header according to Figure 35. Use RV1 to adjust the LCD contrast.

The LCD 1602 is driven by  $I^2C$  by means of the onboard PCF8574 ( $I^2C I/O$  expander) and it does not require any additional wiring.

Signal connections are the same used in the LCD I<sup>2</sup>C adapter of widespread use.

Since ESP32, ESP8266 and NANO33 boards works at 3.3V, to have a sufficient contrast on the display, ICL7660 is used to ensure a sufficient voltage to the V0 pin of the LCD 1602 module.

Depending on component availability, the I/O expander 8574 could be the PCF8574 or the PCF8574A. Both versions perform the same, the only difference is the base address of the I<sup>2</sup>C communication.

Since A0, A1 and A2 of the PCF8574 connect to ground by default (by means of JP17, JP16 and JP15), the default address is set to 0x20 (PCF8574) or 0x38 (PCF8574A).

Use the sample sketches (see § 17.2) to find out which version is your one.



Figure 34 – LCD 16x2 header of the IOT PROTO SHIELD PLUS



Figure 35 – An LCD 1602 display module plugged into the 16 pins header of the IDT PROTO SHIELD PLUS



Figure 36 – Schematic of the IOT PROTO SHIELD PLUS LCD 16x2 section

#### 17.1. Default connections

Refer to § 14.1 to see default routing of I<sup>2</sup>C signals on the **IOT PROTO SHIELD PLUS.** 

#### 17.2. Sample Sketches

Location: IoTPSP\_Sample\_Sketches\IoTPSP\_LCD16x2

These sketches make use of the *LiquidCrystal\_I2C library*, you will find the library in the IoTPSP Sample Sketches\IoTPSP LCD16x2 path.

These sketches use by default the I2C default pins shown in § 14.1.

Use the IoTPSP\_LCD16x2\_NANO sketch to test the LCD1602 with ARDUINO NANO boards. Use the IoTPSP\_LCD16x2\_ESP32 sketch to test the LCD1602 with ESP32 boards. Use the IoTPSP\_LCD16x2\_ESP8266 sketch to test the LCD1602 with ESP8266 boards.

#### 17.3. Changing the I<sup>2</sup>C address of the PCF8574

It is not very usual, but in case that you want / need to change the address of the 8574 chip, you have to OPEN or CLOSE JP17, JP16 and JP15 according to your needs and the PCF8574 datasheet. To OPEN JP17, JP16 or JP15 cut them in the middle. To CLOSE them solder a little drop of tin.

#### 17.4. Using the PCF8574 with different I<sup>2</sup>C GPIO pins

Refer to § 14.3 to see how to use I<sup>2</sup>C with different GPIO pins on the IOT PROTO SHIELD PLUS.

## 18. PCF8574 I2C I/O expander

If you do not need to use the LCD 1602, you can use the on-board 8574 as a general-purpose I/O expander. All the I/O pins of the 8574 are available on the right side of the IDT PROTO SHIELD PLUS (see Figure 37).



Figure 37 – The 8574 pins used for general-purpose I/O

#### **18.1.** Sample Sketches

Location: IoTPSP\_Sample\_Sketches\IoTPSP\_PCF8574

These sketches use by default the I2C default pins shown in § 14.1. These sketches use the *PCF8574 library*, be sure to install it before using the sketches.

orpsp_ssd1306	DISPLAY_128x64_NANO   Ard	uino 1.8.13		
File Edit Sketch To	ols Help	kee		
	Auto Format	Ctrl+T		
	Archive Sketch			
IoTPSP_SSD1	Fix Encoding & Reload			
/********	Manage Libraries	Ctrl+Shift+I		
This is an the sketch m	Serial Monitor	Ctrl+Shift+M		
💿 Library Manager	r		>	×
Type All	V Topic All	√ 8574		
			Version 1.2.0 Version 1.2.0	•
Arduino library More info	for PCF8574 - 8 channel I20	C IO expander Implements shift ro	tate and toggle.	
by Renzo Mischi Arduino, ESP82: read write digita More info	anti 66, smt32 and esp32 librar I values with only 2 wire. Ve	y for PCF8574 i2c digital expander ry simple to use and encoder supp	for Arduino, esp32, SMT32 and ESP8266. Can port.	
by Ricardo Lima Control the PCF8 control the PCF8	Caratti, pu2clr@gmail.com 8574 8-Bit I/O Expander w 574.	ith Serial Interface with your Ardu	uino This library provides an easier interface to	,
			Close	

## 18.2. IoTPSP\_PCF8574\_GET\_ADDRESS

As said before, depending on component availability, the I/O expander 8574 could be the PCF8574 or the PCF8574A.

Both versions perform the same, the only difference is the base address of the I<sup>2</sup>C communication. Since A0, A1 and A2 of the PCF8574 connect to ground by default (by means of JP17, JP16 and JP15), the default address is set to 0x20 (PCF8574) or 0x38 (PCF8574A).

If you have an LCD 1602 you can get the address of the PCF8574 using sketches shown in § 17 (the address of your PCF8574 is shown on the display).

If you do not have an LCD 1602, then you can use the IOTPSP\_PCF8574\_GET\_ADDRESS sketch. The sketch works with ESP32, ESP8266 and NANO boards.

The sketch looks for addresses between 0x20 to 0x27 (PCF8574) or between 0x38 to 0x3F (PCF8574A) and print them on the serial monitor.

Experiment with the examples provided with the *PCF8574 library*.

	нар			_	_	
New Ctrl+	N					Ð.
Open Ctrl+	0					
Open Recent	>					Gent
Sketchbook	>					
Examples	1					
Close Ctrl+	W 11.Arduino	SP >				
Save Ctrl+	S Examples for	ar any board	-o-excander-fast-easy-usage/			
Save As Ctrl+	Shift+S Adafruit Cir	cuit Playground				
D	Bridge	>				
Page setup Ctri+	Ethernet	>				
Print Curi-	Firmata	>				
Preferences Ctrl+	Comma LiquidCryst	al >				
0.0	0 50	>				
La accopy	Stepper	>				
	Temboo	>				
Serial.begin(115.	RETIRED	5				
antaj (1000//	and the second s					
// Set pinMode to	COTPOT Examples fo	ar Arduino Nano Every				
pcf8574.pinMode ()	O, OUTE EEPROM	>				
pcf8574.pinMode(	SoftwareSe	ial >				
Serial.print("In	t pof85 SPI		And use off, and off, the one With interest			
if (pcf8574.begin	()){ Wire		histospiso			
Serial.printin	("OK");	on Custon Libraries	considerit6th Parist ibran Sunction			
Serial nrintin	L"RO"L: ACROPOTO	C SSD1206	coseder//fithEullibroadEuroction			
)	ACKUBUIT	- 3501500	encoderwithruitcioralyrunction			
	Adamut Bu	50	interruptiviemos			
	Adatruit GF	A Library	keyeressed_withinterrupt			
10 100D()	Adafruit SS	01300	keyPressedPin1			
pcf8574.digitalW	Adafruit Ur	ified Sensor	keyPressedPin1_storedOnBuffer_async			
delay(1000);	DFRobotDF	PlayerMini	ledEsp32OnTheSecondI2C			
pcf8574.digitalW	nte (PO, DHT sensor	library	ledWemos			
deray(1000);	Grove 4-Dig	jit Display	readAll_Interrupt			
	LiquidCryst	al I2C	readAll_interrupt_lowMemory			
	PCF8574 lib	rapy	writeAIIEsp8266			
	RICID	3				
	Servo	>		_		_
	StepperDriv	er 🤇 🤇				
	TinyWireM	5				
	TM1637	>				
	INCOMPAT	IBLE >				
		$\nabla$		2002 202		

## 19. TM1637 Digital display module

The **IOT PROTO SHIELD PLUS** is ready to work with 4 pins TM1367 seven segment digital display modules of widespread use.

To work with an TM1367 seven segment digital display, just plug the module into the J20 header according to Figure 40.



Figure 38 – J20 TM1367 digital display header of the IDT PROTO SHIELD PLUS



Figure 39 – Schematic of the IOT PROTO SHIELD PLUS J20 TM1367 digital display header



Figure 40 – An TM1367 digital display module plugged into J20 header of the IDT PROTO SHIELD PLUS

#### **19.1.** Default connections

As shown in Figure 39, the TM1637 section of the **IOT PROTO SHIELD PLUB** uses the I<sup>2</sup>C routing to drive the TM1637 chip.

In some cases, this crate a conflict if you are using other I<sup>2</sup>C devices combined with the TM1637. Refer to § 14.1 to see default routing of I<sup>2</sup>C signals on the **IDT PROTO SHIELD PLUS**.

#### **19.2.** Sample Sketches

Location: IoTPSP Sample Sketches\IoTPSP TM1637 Display

These sketches are based on the tutorial at the following link: <u>https://create.arduino.cc/projecthub/ryanchan/tm1637-digit-display-arduino-quick-tutorial-ca8a93</u>

The tutorial has been modified in order to match the **IOT PROTO SHIELD PLUS** default routing of the I2C signals shown in § 14.1.

The sketches make use of the TM1637 and the Grove 4-Digit Display libraries.

Be sure to install them before using the sketches.



## www.GtronicsShop.com www.Gtronics.NET

Library Mar	nager						
pe All	√ Top	pic All	~	1637			
implemente a bombtime <u>More info</u>	end. On top er(), nightrid	of these regul ler() and boun	ar function cingBall()	ality a segerate fun class v nethod can be used when	hich adds more features using the fun class.	; can be used. For example	2,
M1637 by Avishay Driver for 4 They comm use of any s display moo More info	Orpaz Vers digit 7-seg unicate with special hard dule).	ion 1.2.0 INST ment display n the processor ware (other the	<b>ALLED</b> nodules, b in I2C-lik an GPIO p	<b>ised on the TM1637 chip.</b> i protocol. The implement ns). It is assumed that pu	These chips can be found ation is pure software em Il-up resistors are preser	l in cheap display modules iulation and doesn't make it (usually integrated in the	2
oy AKJ Feature-ful one functior displayTu More info	l and simple nNonblock nable bright	• TM1637 librar ing animation tnessScreen	<b>y with non</b> : Blink, scr clearing a	blocking animation suppor JILleft, fadeIn, fadeOut -C nd splitting, Visit https://gi	t Features: -Display inte ustomizable offset, pad, thub.com/AKJ7/TM1637	gers, float, string with only overflowCustom raw val for more information.	ue Closs
Library Mar	nager						
oe All	√ Top	pic All	~	grove 4-digit			
ability to de More info	tect UV-light	t, visible light i	and intrare	J light. This device is base	d on SII145, a new sens	or from SiLabs.	1
irove 4-Digi by Seeed St Arduino libr gadget, we Seeeduino t can be a nic More info	t Display audio ary to contr utilize a TM: co control the control the choice.	rol Grove_4Dig 1637 to scale o e content, eve	i <b>tal_Displi</b> down the c n the lumi	n <b>y TM1637.</b> 4 digit display i introlling pins into 2 Grove iance of this display. For p	module is usually a 12 pi pins. It only takes 2 dig rojects that require of alp	in module. In this Grove ital pins of Arduino or oha-numeric display, this Install	
						-0	_

			Close		
Use the IOTPSP	TM1637	Display	NANO sketch to test the	TM1637 display	with ARDUIN

Use the IoTPSP\_TM1637\_Display\_NANO sketch to test the TM1637 display with ARDUINO NANO boards.

Use the <code>IoTPSP\_TM1637\_Display\_ESP32</code> sketch to test the TM1637 display with ESP32 boards. Use the <code>IoTPSP\_TM1637\_Display\_ESP8266</code> sketch to test the TM1637 display with ESP8266 boards.

#### 19.3. Using the TM1637 display with different GPIO pins

Refer to § 14.3 to see how to use I<sup>2</sup>C with different GPIO pins on the IOT PROTO SHIELD PLUS.

IOT PROTO SHIELD PLUS USER MANUAL

## 20. RELAY

The **IDT PROTO SHIELD PLUS** is ready to work with 3 pins relay modules of widespread use. To work with a relay module, just plug the module into the J27 header.



Figure 41 - RELAY section of the IDT PROTO SHIELD PLUS



Figure 42 – Schematic of the IDT PROTO SHIELD PLUS RELAY section



Figure 43 – A Relay module plugged into J27 header of the IOT PROTO SHIELD PLUS

#### 20.1. Default connections

	ESP32	ARDUINO	ESP8266
RELAY	GPIO12	D8	NC

*NC* = Not Connected

Please note that there are no default connections for the ESP8266. If you need to use the RELAY module with ESP8266 boards, you need to route the signals with jumper wires.

#### 20.2. Suggested ESP8266 connections

The sample sketch is made to work with this connection: ESP8622 GPIO 0 (D3) connected to header J3 ESP32 GPIO12 (RELAY). ESP8622 GPIO 14 (D5) connected to header J3 ESP32 GPIO32 (TOUCH).



Figure 44 – Suggested connections to work with RELAY and ESP8266

#### 20.3. Sample Sketches

Location: IoTPSP Sample Sketches\IoTPSP RELAY

Use the IoTPSP\_RELAY\_NANO sketch to test the RELAY with ARDUINO NANO boards. Use the IoTPSP\_RELAY\_ESP32 sketch to test the RELAY with ESP32 boards. Use the IoTPSP\_RELAY\_ESP8266 sketch to test the RELAY with ESP8266 boards.

These sketches use the touch sensor to switch the RELAY on. Put your finger on the touch circles, the RELAY will act, and the GRN LED will turn on according to your touch.

See § 13 for more details about TOUCH sensor on the IDT PROTO SHIELD PLUS.

#### 20.4. Using RELAY with a different GPIO pin

According to Figure 42, RELAY connects to the default GPIO pin by means of JP51. To disconnect RELAY from its default GPIO pin, you must OPEN JP51 (cut it in the middle). By means of a jumper wire, connect J25 to another GPIO pin according to your needs.

IOT PROTO SHIELD PLUS USER MANUAL

Please check the default signal routing of the **IDT PROTO SHIELD PLUS** (see §8) to avoid conflict with other default connections.

## 21. DHTxx (humidity and temperature sensor)

The **IOT PROTO SHIELD PLUS** is ready to work with three pins DHT22 or DHT11 modules or "naked" DHT22 or DHT11 sensors of widespread use (see Figure 45).

To work with a DHT module or a naked DHT sensor, just plug it into the J31 header (see Figure 46, Figure 48 and Figure 49).

The DHT naked sensor has four pins. Pin #3 is Not Connected, but if you connect pin#3 to GND it works as well.

So, if you are working with a naked DHT sensor, you can plug it in two different ways: with pin #3 connected to GND or with bent pin #3 (so it will be Not Connected).



Figure 45 – DHT22 three pins module (on the left), Naked DHT22 sensor (on the right)

	00 en 00	Vin-	2000 State		USB 2596 BAT 135		CON R Shill			Yout +	0	Ve Ve Mr1 Gud Hr2 Hr2 Hr2	O 122 124 100 100 100			o (	VINCE NO.	n- * 0 C 1 1	1136	рр 191 006 S	10 10 11 11 11 0 0 11 11 0 0	(an) up (	P31 40DU 900 Vmot	Vout LE Vout J 220 J3	F 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	0 5 010 15	0 1203c 111 115 c		1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1		· · · · · · · · · · · · · · · ·				21 40 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1		4010 日 日 日 日 日 日 日 日 日 日 日 日 日 日 日 日 日 日	IN THE REPORT	1193 1193 1193 1192 1192 1192 1192 1192	ANAL BELL	Contract Rep 2 v c sta
2 (2) •C= H3 •C= C9 129 over																				::::			:																 		A CALLER OF
Gen Detri 131 DHT BRD6																							::::																 		
0 0 0 0 142 0 0 851				::		::	:		:			:	 	:	 :		*					::	*			* ;	::	::		:		::	:	:	 			 			

Figure 46 - DHT section of the IOT PROTO SHIELD PLUS

## www.GtronicsShop.com www.Gtronics.NET



Figure 47 - Schematic of the IOT PROTO SHIELD PLUE DHT section



Figure 48 – A DTH22 module plugged into J31 of the IDT PROTO SHIELD PLUS



Figure 49 – A Naked DTH22 module plugged into J31 of the IDT PROTO SHIELD PLUS

#### 21.1. Default connections

	ESP32	ARDUINO	ESP8266
DHT	GPIO13	D9	NC

*NC* = Not Connected

Please note that there are no default connections for the ESP8266. If you need to use the DHT with ESP8266 boards, you need to route the signals with jumper wires.

#### **21.2.** Suggested ESP8266 connections

The sample sketch is made to work with this connection: ESP8622 GPIO 0 (D3) connected to header J3 ESP32 GPIO13 (DHT).



# 21.3. Sample Sketches

Location: IoTPSP\_Sample\_Sketches\IoTPSP\_DHT These sketches work out of the box with a DHT22 module or DHT22 naked sensor.

#### 21.4. IoTPSP\_DHT\_NANO

The IOTPSP\_DHT\_NANO uses the DHT sensor library, be sure to download it before using the sketch.

ioTPSP_SSD1306	DISPLAY_128x64_NANO   Arduino 1	.8.13	
	Auto Format Archive Sketch Fix Encoding & Reload	Ctrl+T	
/*************************************	Manage Libraries N Serial Monitor	Ctrl+Shift+1 Ctrl+Shift+M	
🥯 Library Mana	ager		×
Type All	V Topic All	√ dht	
Arduino libra These senso <u>More info</u>	ary for AM2320 AM2321 ar rs are similar to DHT12 wit	d AM2323 12C temperature and humidity sensor. Supports AM2320, AM3231, AM232 h I2C interface.	2.
DHT sensor lii by Adafruit Arduino libra Sensors <u>More info</u>	brary ary for DHT11, DHT22, etc	Temp & Humidity Sensors Arduino library for DHT11, DHT22, etc Temp & Humidity           Version 1.4.1          Instal	
by beegee_t Arduino ESP changes: Bae <u>More info</u>	iokyo l <b>ibrary for DHT11, DHT22,</b> ck to working version by re	etc Temp & Humidity Sensors Optimized libray to match ESP32 requirements. Last moving the last commit	
DHT12			~
			Close

Use the <code>IOTPSP\_DHT\_NANO</code> sketch to test the DHT with ARDUINO NANO boards. Open the serial monitor to see the Humidity and Temperature data coming from the DHT sensor.

### 21.5. IoTPSP\_DHT\_ESP32

The IOTPSP\_DHT\_ESP32 uses the DHT sensor library for ESPx, be sure to download it before using the sketch.

IoTPSP_SSD1306_E	)ISPLAY_128x64_NANO   Arduino 1.8.13		
File Edit Sketch Too	ls Help		
	Auto Format	Ctrl+T	
	Archive Sketch		
IoTPSP_SSD1	Fix Encoding & Reload		
/*********	Manage Libraries	Ctrl+Shift+I *	
the sketch m	Serial Monitor	Ctrl+Shift+M	
💿 Library Manager			×
Type All	√ Topic All √ dht		
			^
DHT sensor libra	ry		
by Adafruit			
Sensors	or DHT11, DHT22, etc Temp & Humidit	dity Sensors Arduino library for DHT11, DHT22, etc Temp & Humidity	
**=== :=Z=			
DHT sensor libra	ry for ESPx		
by beegee_toky		unidity Sensors Ontimized libray to match 55022 requirements. Last	
changes: Back t	o working version by removing the last	st commit	
More info			
		Version 1.17.0 Versio	
		4	
by Rob Tillsort			
Arduino library	for I2C DHT12 temperature and humid	idity sensor. DHT12	
More info			¥
		Clos	se

Use the <code>IoTPSP DHT ESP32</code> sketch to test the DHT with ESP32 boards.

Open the serial monitor to see the Humidity and Temperature data coming from the DHT sensor.

#### 21.6. IoTPSP\_DHT\_ESP8266

The IOTPSP\_DHT\_ESP32 uses the DHT sensor library for ESPx, be sure to download it before using the sketch (see § 21.5).

Use the IOTPSP DHT ESP8266 sketch to test the DHT with ESP8266 boards.

Open the serial monitor to see the Humidity and Temperature data coming from the DHT sensor.

#### 21.7. Using DHT with a different GPIO pin

According to Figure 47, DHT connects to the default GPIO pin by means of JP56. To disconnect DHT from its default GPIO pin, you must OPEN JP56 (cut it in the middle). By means of a jumper wire, connect J29 to another GPIO pin according to your needs. Please check the default signal routing of the **IDT PROTO SHIELD PLUS** (see §8) to avoid conflict with other default connections.

## 22. RTC (Real Time Clock)

The **IOT PROTO SHIELD PLUS** is ready to work with I<sup>2</sup>C RTC modules of widespread use (see Figure 51).

The board can work with both RTC DS3231 or DS1307, we suggest DS3231 which is a lot more stable: the time drift is really low even on long period of time.

To work with an RTC module, just plug it into the J11 header.

Since J11 is a four pins header, just check signals on the DS module you are using to be sure to match the J11 pinout (GND, Vcc, SDA and SCL).

Some boards (such as Arduino NANO33 and ESP32) have their own internal RTC, this section is about working with DS3231 or DS1307 RCT modules.



Figure 51 – RTC modules of widespread use, DS3231 (on the left), DS1307 (on the right)



Figure 52 – J11 RTC header of the IOT PROTO SHIELD PLUS



Figure 53 – Schematic of the IOT PROTO SHIELD PLUS J11 RTC header



Figure 54 – An RTC DS3231 module plugged into J11 header of the IDT PROTO SHIELD PLUS

Usually, these modules have a battery holder, and the module is provided with a battery charger circuit. If you are using a rechargeable battery, everything would be ok, but if you are using a NOT rechargeable battery it is suggested to "disable" the charging circuit in order not to damage your battery. The easiest way to "disable" it, is to remove the diode of the charging circuit (see Figure 55).



Figure 55 – The diode of the charging circuit on a DS3231 (on the left), and on a DS1307 (on the right)

#### **22.1.** Default connections

Refer to § 14.1 to see default routing of I<sup>2</sup>C signals on the **IOT PROTO SHIELD PLUS.** 

#### 22.2. Sample Sketches

Location: IoTPSP Sample Sketches\IoTPSP RTC

The sketches make use of the *RTClib* library by Adafruit. Be sure to install it before using the sketches.

IoTPSP_SSD1306_DISPLAY_128x64_NANO   Arduino 1.8.13	
File Edit Sketch Tools Help	
Auto Format Ctrl+T	
Archive Sketch	
IoTPSP_SSD1 Fix Encoding & Reload	
/********** Manage Libraries Ctrl+Shift+1	
the sketch m Serial Monitor Ctrl+Shift+M	
💿 Library Manager	×
Type All V Topic All V rtdib	
D53231	
by Andrew Wickert , Eric Avars, Jean-Claude Wippler, Northern Widget LLC	
Arduino library for the DS3231 real-time clock (RTC) Abstracts functionality for clock reading, clock setting, and alarms for t	the
DS3231 high-precision real-time clock. This is a splice of Ayars' (http://hacks.ayars.org/2011/04/ds3231-real-time-clock.htm and Logher/U-alaber/U-a	ml)
More info	
RTClib	
by Adafruit Version 1.12.4 INSTALLED	
A fork of Jeelab's fantastic RTC library A fork of Jeelab's fantastic RTC library More info	
Select version V Install	
RTCLib by NeiroN	_
by JeeLabs (http://news.jeelabs.org/code/), NeiroN (neiron.nxn@gmail.com)	
A library that makes interfacing DS1302, DS1307, DS3231, PCF8583, PCF8563, RTC_Millis Real Time Clock modules easy.	
Including temperature, alarms and memory storage if present. Includes Date lime class implementation and it conversion. More info	
	~
	Close

Use the IoTPSP\_RTC\_NANO sketch to test the RTC with ARDUINO NANO boards. Use the IoTPSP\_RTC\_ESP32 sketch to test the RTC with ESP32 boards. Use the IoTPSP\_RTC\_ESP8266 sketch to test the RTC with ESP8266 boards.

Open the serial monitor to see date and time red from the RTC module. If you are using a DS3231 module you will get the temperature of the module too, if you are using a DS1307 the temperature data red will be meaningless.

#### 22.3. Using the RTC with different GPIO pins

Refer to § 14.3 to see how to use  $I^2C$  with different GPIO pins on the **IOT PROTO SHIELD PLUS.** 

## 23. SPI on the IoT Proto Shield Plus

SPI (Serial Peripheral Interface) is a synchronous serial communication interface specification used for short-distance communication, primarily in embedded systems (check out more about SPI <u>Wikipedia</u> or other online documentation).



#### **23.1.** Default connections

SPI signals on the **IDT PROTO SHIELD PLUS** are routed by default according to the following connections:

	ESP32	ARDUINO	ESP8266
SPI_MISO	GPIO19	D12	GPIO12 (D6)
SPI_MOSI	GPIO23	D11	GPIO13 (D7)
SPI_CS	GPIO5	D10	GPIO15 (D8)
SPI_SCLK	GPIO18	D13	GPIO14 (D5)

SPI on the **IDT PROTO SHIELD PLUS** is used to work with SD CARD modules (see § 24), or to any other device by means of the SPI1 socket (see Figure 57).



## 24. SD CARD MODULE

The **IOT PROTO SHIELD PLUS** is ready to work with 6 pins SPI SD card modules of widespread use (see Figure 60).

To work with an SPI SD card module, just plug the module into the J10 header (see Figure 60). Of course, you will need a microSD card too.

You may find useful information and knowledges about SD card at the following links:

https://www.arduino.cc/en/reference/SD

https://www.arduino.cc/en/Reference/SDCardNotes

https://create.arduino.cc/projecthub/electropeak/sd-card-module-with-arduino-how-to-read-write-data-37f390



Figure 58 – J10 SD CARD header of the IDT PROTO SHIELD PLUS



Figure 59 – Schematic of the IDT PROTO SHIELD PLUS J10 SD CARD header



Figure 60 – An SPI SD CARD module (on the left) and the module plugged into J10 header of the IOT PROTO SHIELD PLUE (on the right)

#### 24.1. Default connections

Refer to § 23.1 to see default routing of SPI signals on the IDT PROTO SHIELD PLUS.

#### 24.2. Sample Sketches

Location: IoTPSP Sample Sketches\IoTPSP SD CARD

These sketches make use of the SD library, included in the Arduino IDE. These sketches use by default SPI routing shown in § 23.1.

#### 24.3. Testing the SD CARD

These sketches are very useful for testing a card when you're not sure if it's working or not.

Use the IoTPSP\_SD\_CARD\_INFO\_NANO sketch to get SD CARD info with ARDUINO NANO boards. Use the IoTPSP\_SD\_CARD\_INFO\_ESP32 sketch to get SD CARD info with ESP32 boards. Use the IoTPSP\_SD\_CARD\_INFO\_ESP8266 sketch to get SD CARD info with ESP8266 boards.

Open the serial monitor to see info form your SD CARD. If your serial monitor is blank, press the reset button.

#### 24.4. Reading and writing from and to the SD CARD

Once you tested your SD CARD and you are sure it is working (see § 24.3), you can read and write to your SD CARD with the following sketches.

Use the <code>IOTPSP\_SD\_CARD\_RW\_NANO</code> sketch to test writing data to SD CARD and read them back on the serial monitor with ARDUINO NANO boards.

## www.GtronicsShop.com www.Gtronics.NET

Use the <code>IoTPSP\_SD\_CARD\_RW\_ESP32</code> sketch to test writing data to SD CARD and read them back on the serial monitor with ESP32 boards.

Use the <code>IoTPSP\_SD\_CARD\_RW\_ESP8266</code> sketch to test writing data to SD CARD and read them back on the serial monitor with ESP8266 boards.

Open the serial monitor to see data red from your SD CARD. If your serial monitor is blank, press the reset button.

## 25. LDR/xTC

The IOT PROTO SHIELD PLUS is ready to work with LDR, PTC or NTC sensors.

LDR (light dependent resistor, also known as photoresistor) are light sensitive devices to measure the light intensity (more info on <u>Wikipedia</u>).

NTC and PTC (also knowns as thermistor) are resistors whose resistance is strongly dependent on temperature.

NTC are negative temperature coefficient sensors, while PTC are positive temperature coefficient sensors (more info on <u>Wikipedia</u>).

To work with one of these sensors, just plug it into J26 header.

Since these sensors act like a variable resistor the principle is the same you have with a voltage divider (POT).



Figure 61 - LDR/xTC section of the IOT PROTO SHIELD PLUS



Figure 62 – Schematic of the IDT PROTO SHIELD PLUS LDR/xTC Section

#### 25.1. Default connections

	ESP32	ARDUINO	ESP8266
LDR/x1	C GPIO34	A2	NC

*NC* = Not Connected

Please note that there are no default connections for the ESP8266. If you need to use the LDR/xTC with ESP8266 boards, you need to route the signals with jumper wires.

#### 25.2. Suggested ESP8266 connections

The sample sketches are made to work with these connections:

ESP8622 A0 connected to header J3 ESP32 GPIO34 (LDR/xTC).

Remember that ESP8266 A0 pin connects by default to POT (see §11.1). You must open JP12 (see Figure 20) to disconnect ESP8266 A0 pin from POT and let LDR/xTC works properly.

														Γ		٦						
13																			ł		╏	
J5		•																				
ESP32	4	5	12	13	14	15	16	17	18	19	21	22	23	25	26	27	32	33	34	35	36	39
NANO	D5	010	<b>D8</b>	60	90	07	8	01	013	012	A4	A5	011	D4	03	D2	AO	A1	A2	A3	<b>A6</b>	A7
SP8266		15					2	0	14	12	4	ŝ	5		16						AO	

Figure 63 – Suggested connections to work with LDR/xTC and ESP8266

#### 25.3. Sample Sketches

Location: IoTPSP Sample Sketches\IoTPSP LDR xTC

According to the board you are using and the signals routing, the sketches read the analog value of the LDR/xTC and print it to the serial monitor.

Use the IoTPSP\_LDR\_xTC\_NANO sketch to test LDR/xTC with ARDUINO NANO boards. Use the IoTPSP\_LDR\_xTC\_ESP32 sketch to test LDR/xTC with ESP32 boards. Use the IoTPSP\_LDR\_xTC\_ESP8266 sketch to test LDR/xTC with ESP8266 boards.

#### 25.4. Using the LDR/xTC with a different GPIO pin

According to Figure 62, LDR/xTC is routed to the default GPIO pin by means of JP50. To disconnect LDR/xTC from its default GPIO pin, you must OPEN JP50 (cut it in the middle). By means of some jumper wires, connect J24 to another GPIO pin according to your needs. Please check the default signal routing of the IOT PROTO SHIELD PLUS to avoid conflict with other default connections.

## 26. Serial communication on the IOT PROTO SHIELD PLUS

Additional serial communication can be used to communicate with other devices.



Figure 64 – Additional serial communication routing on the IOT PROTO SHIELD PLUS



Figure 65 – J4 header on the IOT PROTO SHIELD PLUS

#### 26.1. Default connections

Additional RX and TX signals on the **IDT PROTO SHIELD PLUB** are routed by default according to the following connections:

	ESP32	ARDUINO	ESP8266
U_RX	GPIO16	D0	GPIO2 (D4)
U_TX	GPIO17	D1	GPIO0 (D3)

On the **IDT PROTO SHIELD PLUS**, this additional serial communication is used by default to communicate with the DFPlayer Mini MP3 (see §27), or to any other device by means of J4 header (see Figure 65).

## 27. DFPlayer Mini MP3 Player

The **IDT PROTO SHIELD PLUB** is ready to work with 8 pins DFPlayer Mini MP3 modules of widespread use (see Figure 66).

To work with a DFPlayer Mini MP3 module, just plug the module into the BRD3 socket (see Figure 66). You can find additional information about the module at the following link:

https://wiki.dfrobot.com/DFPlayer Mini SKU DFR0299

To work with the DFPlayer Mni MP3, you need a microSD card too (with FAT32 file system). The MP3 files need to be into a folder named "mp3".



Figure 66 – A DFPlayer Mini MP3 module (on the left) and the module plugged into BRD3 socket of the Iot Proto Shield Plus (on the right)



Figure 67 – BRD3 socket of the IOT PROTO SHIELD PLUS

+57 +373 JP45 BRD3 DFR0299 PWR .FLAG vec  $\times$  16 BUSY DAC\_R 5 5 DAC\_L R26  $\times \frac{14}{}$ JP46 USB+ 1К 15 USB-2 RY LS1 / 12 ADKEY1 JE Speaker × 13 9 ADKEY2 101 R27 11 JP58 6 SPK1 в SPK2 GND JP59 10 Speak er 01x02 GND J32 GND 123 П JP49 06 **0**<sub>2</sub> BS5138 -1

Figure 68 – Schematic of the IOT PROTO SHIELD PLUS BRD3 socket header

#### 27.1. Default connections

GND

On the **IDT PROTO SHIELD PLUB**, the communication with the DFPlayer Mini MP3 is performed with serial communication.

Refer to §26.1 to see default routing of additional serial communication signals on the IDT PROTO SHIELD PLUS.

The DFPlayer Mini MP3 can work @3V3 as well, but in most cases the 3V3 generated by the board in use (ESP32, NANO, etc.) cannot provide enough current to let the MP3 player works correctly.

Thus, as shown in Figure 68, *Vcc* pin of the DFPlayer Mini MP3 connects by default to 5V (which are usually supplied by the USB port of your computer connected to the board in use).

#### 27.2. Sample Sketches

Location: IoTPSP Sample Sketches \ IoTPSP MP3 Player

These sketches make use of the *DFRobotDFPlayerMini library* by DFRobot. Be sure to install it before using the sketches.

R28 10K

## www.GtronicsShop.com www.Gtronics.NET

ΙσΤ					
PROTO SHIELD PLUS					
USER MANUAL					

	pols Help	1.8.13	
	Auto Format	Ctrl+T	
	Archive Sketch		
IoTPSP_SSD1	Fix Encoding & Reload		
/*****	Manage Libraries	Ctrl+Shift+I	
This is an the sketch m	Serial Monitor	Ctrl+Shift+M	
Library Manager			>
Type All	✓ Topic All ✓ dfp	layer	
			^
Arduino library t More info	o interface with the DFPlayerMini f	MP3 module Arduino library to interface with the DFPlayerMini MP3 mo	odule
by DFRobot Driver for DFPla More info	y <b>er Mini from DFRobot</b> Easy-to-use	e and reliable library for DFPlayer Mini Version 1.0.5 🗸 🏼 In	
			stall
T.II Displaye			
by Gerald Lechne	2F ut data, time and concor values as	speech. The library user the MP2 player module DEDlayer Mini for o	stall
by Gerald Lechno A library to outp The communicat	er ut date, time and sensor values as ion with module is handled in the li	<b>speech.</b> The library uses the MP3 player module DFPlayer Mini for ou braray. No extra driver for DFPlayer is required. Only the pointer on a	utput.

The IoTPSP\_Sample\_Sketches\IoTPSP\_MP3\_Player folder contains also a folder with 3 sample MP3 files. Copy this folder on your micro-SD card to test the IoT PROTO SHIELD PLUS with this samples.

These sketches use by default additional serial communication routing shown in §26.1

#### 27.2.1. IoTPSP\_MP3\_NANO

The IOTPSP\_MP3\_NANO uses the standard serial communication of Arduino NANO boards (routed on D0, D1) to communicate with the DFPlayer Mini MP3.

Of course, since the serial port is used to communicate with the MP3 Player it cannot be used to work with the Serial Monitor.

#### 27.2.2. IoTPSP\_MP3\_NANO\_EVERY\_33

The IoTPSP\_MP3\_NANO\_EVERY\_33 uses the secondary serial communication port on the NANO EVERY or NANO33 boards (routed on D0, D1) to communicate with the DFPlayer Mini MP3.

#### 27.2.3. IoTPSP\_MP3\_ESP32

The IOTPSP\_MP3\_ESP32 uses the secondary serial communication port on the ESP32 boards (routed on GPIO16, GPIO17) to communicate with the DFPlayer Mini MP3.

#### 27.2.4. IoTPSP\_MP3\_ESP8266

The IoTPSP\_MP3\_ESP8266\_SoftwareSerial uses the *EspSoftwareSerial library*, be sure to download it before using the sketch.

IoTPSP_SSD1306_	DISPLAY_128x64_NANO   Ardu	uino 1.8.13			
File Edit Sketch Too	ols Help				
	Auto Format	Ctrl+T			
	Archive Sketch				
IoTPSP_SSD1	Fix Encoding & Reload				
/**********	Manage Libraries	Ctrl+Sh	ift+l		
the sketch m	Serial Monitor	Ctrl+Sh	<sup>ift+M</sup> 77		
🥯 Library Manager					×
Type All	√ Topic All	✓ EspSoftwareSerial			
EspSoftwareSeria	al				^
by Dirk Kaar, Pe	ter Lerup	arial for ESD8266/ESD3	2		
More info	for the Arduno software s				
				Version 6.11.3 V	Install
					45
					~
					Close

The IoTPSP\_MP3\_ESP8266\_SoftwareSerial create an additional software serial port on GPIO0 and GPIO2 to communicate with the DFPlayer Mini MP3.

The ESP8266 has a secondary serial communication port routed on GPIO1 and GPIO3, in case you want to use this second serial port, you need to reroute the signals and experimenting by yourself.

#### 27.3. Using the DFPlayer Mini MP3 module with a different GPIO pins

It is not recommended to use different pins to establish a serial communication with the DFPlayer Mini MP3 module.

Anyway, if you are using an "old" Arduino NANO board you may want to use the Serial Monitor, thus you need to free D0 and D1.

In this case we will use the *SoftwareSerial library* to create a "virtual" serial port.

According to Figure 68, U\_RX and U\_TX connects by default to TX and RX pins of the DFPlayer Mini MP3 module by means of JP46 and JP47.

You can experiment by OPENING JP46 and JP47 (cut them in the middle), rerouting RX and TX signals via J21 and then using the sketch described in § 27.4.

## 27.4. IoTPSP\_MP3\_NANO\_SoftwareSerial

The <code>IoTPSP\_MP3\_NANO\_SoftwareSerial</code> uses the *SoftwareSerial library*, included in the Arduino IDE.

To use this sketch, refer to § 27.3. The sketches uses D10 as RX pin and D11 as TX pin. Additional information about Software Serial can be found at this link <u>https://www.arduino.cc/en/Reference/softwareSerial</u>.

## **28. SPEAKER**

The SPEAKER can be used also as a buzzer or to create frequency modulated tones. It does not come "out of the box" and two jumper wires are required. To let the SPEAKER work as a tone buzzer you have to:

- Remove the DFPlayer Mini MP3 module from the BRD3 socket
- Use a jumper wire to connect BRD3 pin #1 (Vcc) to BRD3 pin #6 (SPK1)
- Use a wire to connect J21 pin #2 to J23



Figure 69 – Schematic of the IOT PROTO SHIELD PLUB SPEAKER buzzer configuration



#### Figure 70 – Suggested wiring to work with SPEAKER as a tone BUZZER

#### 28.1. Default connections

According to Figure 69, U\_TX will drive the buzzer tone. As shown in § 26.1, U\_TX connects by default to the following pins:

	ESP32	ARDUINO	ESP8266		
U_TX	GPIO17	D1	GPIO0 (D3)		

#### 28.2. Sample Sketches

Location: IoTPSP Sample Sketches \ IoTPSP SPEAKER

#### 28.3. IoTPSP\_SPEAKER\_NANO

The IoTPSP\_SPEAKER\_NANO is based on the *tone()* function of the Arduino base library. Information about the Arduino *tone()* function can be found at this link: <a href="https://www.arduino.cc/reference/en/language/functions/advanced-io/tone/">https://www.arduino.cc/reference/en/language/functions/advanced-io/tone/</a> The function does not work on NANO33 BLE (at least at the moment of writing this manual).

#### 28.4. IoTPSP\_SPEAKER\_ESP32

The IoTPSP\_SPEAKER\_ESP32 uses the *Tone32* library. The library is downloadable at <u>https://github.com/lbernstone/Tone32</u>. A copy of the library is also contained in the sketch folder.

#### Rev.1.07

#### 28.5. IoTPSP\_SPEAKER\_ESP8266

The IoTPSP\_SPEAKER\_ESP8266 is based on the *tone()* function of the Arduino base library. Information about the Arduino *tone()* function can be found at this link: <u>https://www.arduino.cc/reference/en/language/functions/advanced-io/tone/</u>

#### 29. MIC

The **IOT PROTO SHIELD PLUS** is ready to work with MAX9814 or MAX4466 microphone modules of widespread use.



Figure 71 – Schematic of the IOT PROTO SHIELD PLUS MIC Section



Figure 72 - MIC section of the IOT PROTO SHIELD PLUS
#### **29.1.** MAX9814 microphone module

This microphone module is a 5 pins module based on MAX9814 chip. More details on the MAX9814 chip at this link: <u>https://datasheets.maximintegrated.com/en/ds/MAX9814.pdf</u>

This module has 5 pins: Vdd: Module power supply – 2.7-5.5 V GND: Ground OUT: Analog data output to the microcontroller Gain: Adjusting maximum module output\* AR: Adjusting module accuracy\*

*	Float	Vcc	GND
Gain	60dB	50dB	40dB
AR	1:4000	1:2000	1:500

The module's output is not zero in silent mode and has an offset 1.25V.



Figure 73 – A MAX9814 microphone module of widespread use

To work with a MAX 9814 microphone module just plug it into J19 as shown in the following picture.



Figure 74 – A MAX9814 microphone plugged into J19

## 29.2. Setting the Gain

By default, Gain is set to 40dB: J19 pin #3 connects to GND by means of JP48. Opening JP48 (cut it in the middle) left J19 pin #3 floating, thus gain is set to 60dB. With JP48 open and J22 pin 3# connected to Vcc, gain is set to 50dB.

## 29.3. Setting the module accuracy

By default, AR (adjusting module accuracy) is set to 1:4000: J19 pin #1 is floating. Connecting J22 pin #1 to Vcc sets AR to 1:2000. Connecting J22 pin #1 to GND sets AR to 1:500.

## 29.4. MAX4466 microphone module

This module is a 3 pins module based on MAX4466 chip. More details on the MAX4466 chip at this link: <u>https://www.maximintegrated.com/en/products/analog/audio/MAX4466.html</u>



Figure 75 – A MAX4466 microphone module of widespread use

To work with a MAX4466 microphone module plug it into J19 as shown in the following picture. Be sure that JP48 is closed: JP19 pin #3 must connect to GND.



Figure 76 – A MAX4466 microphone plugged into J19

#### 29.5. Default connections

	ESP32	ARDUINO	ESP8266
MIC	GPI39	A7	NC

*NC* = Not Connected

Please note that there are no default connections for the ESP8266. If you need to use the MIC with ESP8266 boards, you need to route the signals with jumper wires.

#### 29.6. Suggested ESP8266 connections

The sample sketches are made to work with these connections:

ESP8622 A0 connected to header J3 ESP32 GPIO39 (MIC).

Remember that ESP8266 pin A0 connects by default to POT (see §11.1). You must OPEN JP12 (see Figure 20) to disconnect ESP8266 pin A0 from POT and let the MIC works properly.

																					I	
73	Γ																				•	ł
J5		•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
ESP32	4	S	12	13	14	15	16	17	18	19	21	22	23	25	26	27	32	33	34	35	36	39
NANO	D5	D10	D8	<b>D</b> 9	D6	D7	00	<b>D1</b>	D13	012	A4	A5	011	D4	D3	D2	AO	A1	A2	A3	A6	A7
SP8266		15					2	0	14	12	4	5	13		16						AO	

Figure 77 – Suggested connections to work with MIC and ESP8266

#### 29.7. Sample Sketches

Location: IoTPSP\_Sample\_Sketches\IoTPSP\_MIC

According to the board you are using and the signals routing, the sketches read the analog value of the MIC and print it to the serial monitor.

Use the IoTPSP\_MIC\_NANO sketch to test the MIC with ARDUINO NANO boards. Use the IoTPSP\_MIC\_ESP32 sketch to test the MIC with ESP32 boards. Use the IoTPSP\_MIC\_ESP8266 sketch to test the MIC with ESP8266 boards.

## 29.8. Using the MIC with a different GPIO pin

According to Figure 71, MIC connects to the default GPIO pin by means of JP28. To disconnect MIC from its default GPIO pin, you must OPEN JP28 (cut it in the middle). By means of some jumper wires, connect J22 pin #2 to another GPIO pin according to your needs. Please check the default signal routing of the IDT PROTO SHIELD PLUS to avoid conflict with other default connections.

IOT PROTO SHIELD PLUS USER MANUAL

## 30. SERVO

The **IOT PROTO SHIELD PLUS** is ready to work with a standard **5V** hobby servo of widespread use. To connect a servo, you need to solder a standard 2,54mm 3 pins single row male header (not provided with the board) to J39.



Figure 78 - SERVO section of the IDT PROTO SHIELD PLUS



Figure 79 - Schematic of the IOT PROTO SHIELD PLUS SERVO Section

#### **30.1.** Default connections

	ESP32	ARDUINO	ESP8266
SERVO	GPIO04	D5	NC

NC = Not Connected

Servo Power Supply connects by default to 5V.

Please note that there are no default connections for the ESP8266. If you need to use the SERVO with ESP8266 boards, you need to route the signals with jumper wires.

### 30.2. Suggested ESP8266 connections

The sample sketch is made to work with this connection: ESP8622 GPIO2 (D4) connected to header J3 ESP32 GPIO04 (SERVO).



Figure 80 – Suggested connections to work with SERVO and ESP8266

#### **30.3.** Sample Sketches

Location: IoTPSP Sample Sketches\IoTPSP SERVO

These sketches use the POT to move the servo position. Rotate the POT knob to change the angular servo position.

See §11 for more details about POT on the IOT PROTO SHIELD PLUS.

#### 30.4. IoTPSP\_SERVO\_NANO

The IoTPSP\_SERVO\_NANO sketch, uses the Servo library.

Be sure to install it before using the sketch.

Be sure to use the latest version, if you are using NANO 33 boards or EVERY (old releases of the library will not work).

More details here <a href="https://www.arduino.cc/reference/en/libraries/servo/">https://www.arduino.cc/reference/en/libraries/servo/</a>

IoTPSP_SSD130	06_DISPLAY_128x64_NANO   Arduino 1.8.1	3
File Edit Sketch	Tools Help	
	Auto Format	Ctrl+T
	Archive Sketch	
IoTPSP_SSD1	Fix Encoding & Reload	
/*****	Manage Libraries	Ctrl+Shift+I
This is an the sketch m	Serial Monitor	Ctrl+Shift+M

Rev.1.07

💿 Library Manager	×
Type All V Topic All V servo	
EduIntro by Arduino LLC Library used for super-fast introduction workshops Is intended to be used with Arduino UNO / MICRO / MEGA / NANO classic / NANO Every / MKR / WiFi REV2 and a set of basic components (led, button, piezo, LM35, thermistor, LDR, PIR, DHT11, and servo) as a way to introduce people to the basic aspects of Arduino during short workshops. More info Version 0.0.13 v Install	^
Servo by Michael Margolis, Arduino Version 1.1.7 INSTALLED Allows Arduino boards to control a variety of servo motors. This library can control a great number of servos. It makes careful use of timers: the library can control 12 servos using only 1 timer. On the Arduino Due you can control up to 60 servos. <u>More info</u>	
Accessories	1
by Thierry Paris - Locoduino This is a library for Arduino to handle accessories like lights, motors. This library can handle coil motors, stepper motors, lights, servos.	~
Clos	æ

Use the <code>IoTPSP\_SERVO\_NANO</code> sketch to test the SERVO with ARDUINO NANO boards.

## 30.5. IoTPSP\_SERVO\_ESP32

The IOTPSP\_SERVO\_ESP32 sketch, uses the *Esp32Servo* library. Be sure to install it before using the sketch. More details here https://www.arduino.cc/reference/en/libraries/esp32servo/

IoTPSP_SSD1306	_DISPLAY_128x64_NANO   Ard	luino 1.8.13				
File Edit Sketch To	pols Help					
	Auto Format		Ctrl+T			
	Archive Sketch					
IoTPSP_SSD1	Fix Encoding & Reload			-		
/**********	Manage Libraries		Ctrl+Shift+I			
the sketch m	Serial Monitor		Ctrl+Shift+M	E		
💿 Library Manage	er					×
Type All	V Topic All	✓ esp32servo				
ESP32Servo						
Allows ESP32 b types of servos It makes use o No attempt has <u>More info</u>	oards to control servo, ton , ,f the ESP32 PWM timers: th ; been made to support mu	e and analogWrit ne library can cont Itiple servos per c	e motors usin rol up to 16 s hannel.	g Arduino semantic: ervos on individual c	s. This library can control a m channels Version 0.9.0 V	any
by Sébastien M Initial develope More info	iatos <b>ment release</b> Control Parall	ax Feedback 360°	' High Speed \$	Servos with a ESP32	dev-board.	
						~
						Close

Use the IOTPSP SERVO ESP32 sketch to test the SERVO with ESP32 boards.

#### 30.6. IoTPSP\_SERVO\_ESP8266

The IOTPSP\_SERVO\_ESP8266 sketch, uses the Servo library (see § 30.4). Be sure to install it before using the sketch. Servo signal must be routed as shown in Figure 80.

Use the <code>IoTPSP\_SERVO\_ESP8266</code> sketch to test the SERVO with ESP8266 boards.

### 30.7. Using SERVO with a different GPIO pin

According to Figure 79, SERVO connects to the default GPIO pin by means of JP53. To disconnect SERVO from its default GPIO pin, you must OPEN JP53 (cut it in the middle). By means of a jumper wire, connect J30 pin #1 to another GPIO pin according to your needs. Please check the default signal routing of the IDT PROTO SHIELD PLUS to avoid conflict with other default connections.

### **30.8.** Powering the SERVO with a different power supply

According to Figure 79, *Vservo* connects by default to 5V by means of JP54. First thing first you must disconnect *Vservo* from 5V (cut JP54 between pads 1 and 2). **Be sure that JP54 pads 1-2 connection is OPEN** to avoid damage on the **IDT PROTO SHIELD PLUS**. At this point you can reroute *Vservo* according to JP52 and JP54 shown in Figure 79, or you can use J30 pin #2 to connect an external power supply (in case do not forget that GND must be in common).

## 31. HC-SR04 (distance sensor)

The **IOT PROTO SHIELD PLUS** is ready to work with 4 pins HC-SR04 distance sensor modules of widespread use (see Figure 81).

The power supply of the HC-SR04 must be 5V.

The TRIGGER signal (HC\_SR\_T) works @3V3 as well, so it doesn't matter if you are working with a 5V board (i.e. NANO or EVERY).

Since ECHO is a 5V signal, the **IDT PROTO SHIELD PLUB** provides a clipping circuit on this signal in order to let the HC-SR04 works with 3V3 tolerant pin as well (i.e. ESP32, NANo33, etc.) To work with a HC-SR04 module, just plug the module into the J14 header.



Figure 81 – HC-SR04 module of widespread use



Figure 82 – HC-SR04 section of the IOT PROTO SHIELD PLUS



Figure 83 – Schematic of the IOT PROTO SHIELD PLUS HC-SR04 Section



Figure 84 - An HC-SR04 module plugged into J14 header of the IoT PROTO SHIELD PLUS

#### 31.1. Default connections

	ESP32	ARDUINO	ESP8266
HC_SR_T	GPIO14	D6	NC
HC_SR_E	GPIO15	D7	NC

*NC* = Not Connected

Please note that there are no default connections for the ESP8266. If you need to use the HC-SR04 with ESP8266 boards, you need to route the signals with jumper wires.

#### **31.2.** Suggested ESP8266 connections

The sample sketch is made to work with this connection: ESP8622 GPIO 4 (D2) connected to header J3 ESP32 GPIO14 (HC\_SR\_T). ESP8622 GPIO 5 (D1) connected to header J3 ESP32 GPIO15 (HC\_SR\_E).



Figure 85 – Suggested connections to work with HC-SR04 and ESP8266

### **31.3.** Sample Sketches

Location: IoTPSP\_Sample\_Sketches\IoTPSP\_HC\_SR04

Use the <code>IoTPSP\_HC\_SR04\_NANO</code> sketch to test the HC-SR04 with ARDUINO NANO boards. Use the <code>IoTPSP\_HC\_SR04\_ESP32</code> sketch to test the HC-SR04 with ESP32 boards. Use the <code>IoTPSP\_HC\_SR04\_ESP8266</code> sketch to test the HC-SR04 with ESP8266 boards.

Open the serial monitor to see the distance measured by the SR-HC04 sensor.

More details at this link: <u>https://create.arduino.cc/projecthub/Isaac100/getting-started-with-the-hc-sr04-ultrasonic-sensor-036380</u>

## 31.4. Using HC-SR04 with different GPIO pins

According to Figure 83, HC\_ST\_T and HC\_SR\_E connect to their default GPIO pins by means of JP39 and JP40.

To disconnect HC\_ST\_T and HC\_SR\_E from their default GPIO pins, you must OPEN JP39 and JP40 (cut them in the middle).

Solder a 2 pins header on J17.

By means of two jumper wires, connect J17 to other GPIO pins according to your needs.

Please check the default signal routing of the **IOT PROTO SHIELD PLUS** to avoid conflict with other default connections.

## **32. STEPPER MOTOR**

The **IOT PROTO SHIELD PLUS** is ready to work with DRV8825 form factor stepper driver modules. These modules are used to drive 2 phases 4 wires (A, /A, B, /B) stepper motors.

The wiring diagram slightly changes according to the driver you are using (see further in this manual), anyway there are some common minimum wiring requirements whose apply to all drivers:

- Motor power supply
- Logic power supply
- Sleep and (and Reset) wiring
- Step signal wiring
- Direction signal wiring

According to Figure 86 and §32.8, the **IDT PROTO SHIELD PLUS** is ready to work with a DRV8825 without any additional connection (or disconnection).

Consider that *Vmot* is disconnected by default. You have to provide the right motor power supply voltage by means of J12 or according to signal routing shown in Figure 87.

Please read carefully this entire chapter (§32) before using a stepper motor: you must consider all the aspects and all the options provided by the IDT PRDID SHIELD PLUS, to correctly drive a stepper motor.



Figure 86 – Schematic of the IOT PROTO SHIELD PLUS Motor Stepper Section



Figure 87 – Schematic of the Vmot routing section of the IoT PROTO SHIELD PLUB

BRD1 (MT3608 step-up-Module) is a predisposition for possible future features, do not consider it now.



Figure 88 - BRD2 socket of the IOT PROTO SHIELD PLUS



Figure 89 – J15 screw connector of the IDT PROTO SHIELD PLUS

### 32.1. Default connections

	ESP32	ARDUINO	ESP8266
MOT_DIR	GPIO25	D4	NC
MOT_STP	GPIO26	D3	GPIO16 (D0)

NC = Not Connected

www.GtronicsShop.com www.Gtronics.NET

Please note that there are no default connections for the MOT\_DIR signal when working with ESP8266. If you need to use a stepper motor with ESP8266 boards, you need to route MOT\_DIR signal with a jumper wire.

#### 32.2. Suggested ESP8266 connections

The sample sketch is made to work with this connection: ESP8622 GPIO14 (D5) connected to header J3 ESP32 GPIO25 (MOT\_DIR).



Figure 90 – Suggested connections to work with a stepper motor and ESP8266

#### 32.3. Sample Sketches

```
Location: IoTPSP_Sample_Sketches\IoTPSP_STEPPER_MOTOR
```

According to step per revolution of your motor, these sketches make rotating the motor clockwise and counterclockwise in a loop.

## 32.4. IoTPSP\_STEPPER\_MOTOR\_NANO

The IoTPSP\_STEPPER\_MOTOR\_NANO is based on the *StepperDriver* library. Be sure to install it before using the sketch. Information about the Arduino *StepperDriver* can be found at this link: <u>https://www.arduino.cc/reference/en/libraries/stepperdriver/</u> totalson of the second state of



	anager	
oe All	V Topic All	StepperDriver
ighPower oy Pololu ligh-Powe	StepperDriver er Stepper Motor Driver library for	Arduino This is a library for the Arduino IDE that helps interface with Pololu
lign-Powe <u>tore info</u>	r Stepper Motor Drivers.	
epperDri	iver	
y Lauren 4988, DF	tiu Badea Version 1.3.1 INSTALLED RV8825 and generic two-pin steppe	r motor driver library. Control steppers via a driver board providing STEP+DIR like
he ones f RV8834, Acre info	rom Pololu. Microstepping is suppor DRV8880.	ted. Acceleration is supported. Supported drivers are A4988, DRV8824, DRV8825,
Select vers	ion 🗸 Install	
	143	

## 32.5. IoTPSP\_STEPPER\_MOTOR\_ESP32

The IOTPSP\_STEPPER MOTOR\_ESP32 uses the ESP-FlexyStepper library.

Be sure to install it before using the sketch.

More details at the following link: <u>https://www.arduino.cc/reference/en/libraries/esp-flexystepper/</u>

File Edit Sketch To	ols Help		
	Auto Format	Ctrl+T	
	Archive Sketch		
IoTPSP_SSD1	Fix Encoding & Reload		
/*******	Manage Libraries	Ctrl+Shift+I	
This is an the sketch m	Serial Monitor	Ctrl+Shift+M	
the sketch m			
Library Manager			×
Type All	V Topic All V	ESP-flexy	
by Paul Resper V. This library is use stepper motors fin functionality More infe Version 1.4.2	ersion 1.4.3 INSTALLED d to control one or more stepp om an ESP32 device. It is bas	per motors from an ESP32 device This library is used to control one or more ed on the FlexyStepper library by S.Reifels but provides some additional	-
		Clos	se

## 32.6. IoTPSP\_STEPPER\_MOTOR\_ESP8266

The IoTPSP\_STEPPER\_MOTOR\_ESP8266 is based on the "Connecting all control pins from DRV8825 to NodeMCU" from the link below <a href="https://hackaday.io/project/160569-nodemcu-and-drv8825">https://hackaday.io/project/160569-nodemcu-and-drv8825</a>

#### **32.7.** Choosing the Stepper Driver Module

As mentioned above in this manual, there are different motor drivers, they are slightly different and each one has its pros and cons.

Regardless of the motor driver model, the first thing to do when working with these drivers, is to set current per phase according to the motor you are using.

A detailed explanation is shown at this link: https://www.youtube.com/watch?v=89BHS9hfSUk

### 32.8. DRV8825 module

This module is based on the DRV8825 Stepper Motor Controller IC.

It is recommended to read the datasheet (<u>https://www.ti.com/lit/ds/symlink/drv8825.pdf</u>) before using this module.

A more detailed explanation of this module (and in general DRV8825 based modules) can be found here: <u>https://www.pololu.com/product/2133</u>

- DRV8825 module PROS
  - Reasonably cheap
  - Of widespread use
  - Easy to find on the market
  - The IDT PROTO SHIELD PLUS is ready to work with this module without changing any JP configuration (see Figure 86 and Figure 91)
- DRV8825 module CONS
  - It needs 8.2-45V motor power supply, which means that you will need an external power supply to let the motor work



Figure 91 – Minimal wiring diagram drive a stepper motor with a DRV8825 module (full-step mode)

To work with a DRV8825 module on the IOT PROTO SHIELD PLUS:

- According to Figure 95 and Figure 86, JP27 and JP33 must be CLOSED (default configuration), in order to connect RESET and SLEEP pins of the DRV8834 module to Vbrd of the IDT PRDTD SHIELD PLUS
- plug it into BRD2 socket according to the pinout (see Figure 92)
- connect the motor A, /A, B, /B according to motor wiring and Figure 89
- connect motor power supply according to motor specs and Figure 89





Figure 92 – A DRV8825 module plugged into BRD2 socket

#### 32.9. TMC2xxx module

This module is based on the TMC21xx to TMC26xx Stepper Motor Controller IC. It is recommended to read the explanation and datasheet (<u>https://www.trinamic.com</u>) before using this module.

- TMC2xxx module PROS
  - Noiseless current control
  - Becoming of widespread use (at the moment while writing this manual)
  - Easy to find on the market
  - A minimal modification of the IDT PRDTD SHIELD PLUS configuration is required to work with this module (see below)
  - $\circ$   $\;$  It works with 3V to 5V logic power supply
  - It works with 5V to 36V motor power supply (which means that with current per phase lower than 0.5A you can use the 5V provided by the USB of your PC to let the motor work)
- TMC2xxx module CONS
  - A little more expansive compared to a DRV8825

Considering the above-mentioned PROS and CONS, the modules based on TMC chips are the more versatile choice to experiment with stepper motor driver.



To work with an TMC2xxx module on the IDT PROTO SHIELD PLUS:

- according to Figure 95 and Figure 86, JP29 must be CLOSED in order to connect VIO pin of the TMC2xxx module to Vbrd
- plug the TMC2xxx module into BRD2 socket according to the pinout (see Figure 94)
- connect the motor A, /A, B, /B according to motor wiring and Figure 89
- according to the motor specs, Figure 87 and Figure 89, provide the required motor power supply
  - if you want to use 5V coming from the USB of your PC (or laptop) you need to CLOSE JP31 (with a drop of tin)
- this module requires EN pin to be LOW to enable the motion (connect J13 Pin#5 to GND)



Figure 94 – A TMC2xxx module plugged into BRD2 socket

## 32.10. DRV8834 module

This module is based on the DRV8834 Stepper Motor Controller IC.

It is recommended to read the datasheet (<u>https://www.ti.com/lit/ds/symlink/drv8834.pdf</u>) before using this module.

A more detailed explanation of this module (and in general DRV8834 based modules) can be found here: <u>https://www.pololu.com/product/2134</u>

- DRV8834 module PROS
  - Works with a motor power supply range between 2.5 and 10.8V (which means that with current per phase lower than 0.5A you can use the 5V provided by the USB of your PC to let the motor work)
  - A minimal modification of the IDT PRDTD SHIELD PLUS configuration is required to work with this module (see below)
- DRV8834 module CONS
  - o Quite expensive if compared with other modules
  - More difficult to find on the market
  - Not suitable to drive motors whose require a voltage greater than 10.2V (anyway a NEMA17 o similar works fine)



Figure 95 – Minimal wiring diagram drive a stepper motor with a DRV8834 module (full-step mode)

To work with a DRV8834 module on the IDT PROTO SHIELD PLUS:

- according to Figure 95 and Figure 86 JP27 must be CLOSED (default configuration), in order to connect SLEEP pin of the DRV8834 module to Vbrd of the IDT PROTO SHIELD PLUS
- according to Figure 95 and Figure 86 JP33 must be OPEN, in order to disconnect *VREF* pin of the DRV8834 module from Vbrd of the IDT PROTO SHIELD PLUS
- plug the DRV8834 module into BRD2 socket according to the pinout (see Figure 96)
- connect the motor A, /A, B, /B according to motor wiring and Figure 89
- according to the motor specs, Figure 87 and Figure 89, provide the required motor power supply
  - $\circ$   $\,$  if you want to use 5V coming from the USB of your PC you need to CLOSE JP31 (with a drop of tin)



Figure 96 – A DRV8834 module plugged into BRD2 socket

#### 32.11. A4988 module

This module is based on the A4988 Stepper Motor Controller IC.

It is recommended to read the datasheet (<u>https://www.pololu.com/file/0J450/A4988.pdf</u>) before using this module.

A more detailed explanation of this module (and in general A4988 based modules) can be found here: <u>https://www.pololu.com/product/1182</u>

- A4988 module PROS
  - Reasonably cheap
  - Of widespread use

- Easy to find on the market
- A minimal modification of the IDT PRDTD SHIELD PLUS configuration is required to work with this module (see below)
- A4988 module CONS
  - It needs 8-35V motor power supply, which means that you will need an external power supply to let the motor work



Figure 97 – Minimal wiring diagram drive a stepper motor with a A4988 module (full-step mode)

To work with an A4988 module on the IDT PROTO SHIELD PLUS:

- according to Figure 95 and Figure 86, JP27 must be OPEN in order to disconnect RESET and SLEEP pins of the A4988 module from Vbrd of the IDT PROTO SHIELD PLUS
- according to Figure 95 and Figure 86, JP33 must be CLOSED in order to connect RESET and SLEEP pin of the A4988 module
- according to Figure 95 and Figure 86, JP29 must be CLOSED in order to connect VDD pin of the A4988 module to Vbrd
- Plug the A4988 module into BRD2 socket according to the pinout (see Figure 98)
- connect the motor A, /A, B, /B according to motor wiring and Figure 89
- according to the motor specs, Figure 87 and Figure 89, provide the required motor power supply



Figure 98 – An A4988 module plugged into BRD2 socket

### 32.12. Choosing and connecting the motor

Starting from your application you will choose a motor that better fits your needs. According to the motor specifications, you can choose the stepper driver module that better fits your application.

Following are just two examples of different stepper motors for your reference:

- NEMA17 motor of widespread use (see Figure 99)
- micro motor (see Figure 100).



Figure 99 – NEMA 17 stepper motor of widespread use



Figure 100 – Micro stepper motor

## 32.13. Using STEPPER MOTOR with different GPIO pins or use additional

#### features

According to Figure 86, MOT\_DIR and MOT\_STP connect to their default GPIO pins by means of JP37 and JP38.

To disconnect MOT\_DIR and MOT\_STP from their default GPIO pins, you must OPEN JP37 and JP38 (cut them in the middle).

By means of two jumper wires, connect J13 pin #6 and pin #7 to other GPIO pins according to your needs. Looking at Figure 91, Figure 95 and Figure 97 you can see that some pins (such as the ones used to set microsteps) are not connected by default, you can access to the functions provided by those pins by means of J13 referring to the schematic sown in Figure 86.

Please check the default signal routing of the IDT PROTO SHIELD PLUS to avoid conflict with other default connections.

## **33. LEVEL SHIFTER**

Since ESP NANO33 boards works with 3V3 operating voltage (see §4), you may need to convert 5V to 3V3 or vice versa.

You can do this level shifting using the 4 onboard level shifter provided on the IDT PROTO SHIELD PLUS.

The onboard level shifter is a copy of the level shifter module of widespread use.

The level shifter provides HV (HighVoltage) and LV (LowVoltage) connections (see Figure 102).

The level shifter converts an HV voltage signal into LVin voltage signal and, vice versa, it converts a LV voltage signal into an HVin voltage signal.

By default, HVin connects to 5V (by means of JP41) and LVin connects to 3V3 (by means of JP42).





Figure 101 – The Level Shifter section of the IoT PROTO SHIELD PLUS



Figure 102 – Schematic of the Level Shifter section of the IDT PROTO SHIELD PLUS

## 33.1. Using the onboard level shifter with different HVin and / or LVin

If you need to use the onboard level shifter with different HVin, you must disconnect JP41 pad #1 from pad #2 (cut in the middle) and connect JP41 pad #2 to pad #3 (with a little drop of tin). Now you must provide your desired HVin to J16 pin #3 header.



Figure 103 – JP41 and J16 pin#3 of the Level Shifter section of the IoT PROTO SHIELD PLUS

# www.GtronicsShop.com www.Gtronics.NET

IOT PROTO SHIELD PLUS USER MANUAL

If you need to use the onboard level shifter with different LVin you must disconnect JP42 pad #1 from pad #2 (cut in the middle) and connect JP42 pad #2 to pad #3 (with a little drop of tin). Now you must provide your desired LVin to J18 pin #3 header.



Figure 104 – JP42 and J18 pin#3 of the Level Shifter section of the IoT PROTO SHIELD PLUS

## 34. ESP01 WiFi module

If you want to experiment with an ESP01WiFi module (ancestor of the newer ESP8266 modules) you can solder a 2x4 header to BRD6 pads and solder a 1x8 header to J42 (see Figure 105). J42 lets you access ESP01 pins easier than with its 2x4 native footprint.





Figure 105 – The ESP01 section of the IDT PROTO SHIELD PLUS



Figure 106 - Schematic of the ESP01 section of the IDT PROTO SHIELD PLUS

## **35. EXTERNAL POWER SUPPLY**

#### DISCLAIMER

This section is recommended for advanced user, that's why this part is not populated on the board. Refer to the IDT PROTO SHIELD PLUE schematics (see §36) and to the technical specifications of the board you are using to correctly use an external power supply combined with an LM2596 module. Choosing a wrong voltage regulator or configure a wrong routing of Vin power rail (see schematics on §36) may damage the ESP or NANO board you are using.

The **IOT PROTO SHIELD PLUS** can accept typical LM2596 DC-DC step down modules of widespread use (see Figure 107).

The LM2596 module must be plugged into the IoT PROTO SHIELD PLUS according to Figure 109. To plug the LM2596 module you need four 2 pins header (not included in the IoT PROTO SHIELD

**PLUS**) to be soldered to Vin+, Vin-, Vout+ and Vout- pads of the **IDT PROTO SHIELD PLUS**. The external power supply can be provided by means of JP2 screw terminals connector (see Figure 108 and Figure 110).

If you do not have a 2 screw terminals connector, you can use the two pads of the board and solder two wires directly to them.

The 2 pads screw terminals connector is not provided with the **IDT PROTO SHIELD PLUS** and it must be purchased and soldered separately.

The external power supply connects to IN+ (Vin+ on the silkscreen) of the LM2596 module by means of JP57, which must be CLOSED (with a little drop of tin).

If you want to add a reverse polarity protection you can add D4 (i.e., 1N5819, see Figure 110) but in this case JP57 must be OPEN.

The OUT+ (Vout+ on the silkscreen) regulated voltage of the LM2596 module connects to Vin of the IDT **PROTO SHIELD PLUS**.





Figure 107 – A typical LM2596 DC-DC module of widespread use



Figure 108 – The External power supply section of the IOT PROTO SHIELD PLUS



Figure 109 – A typical LM2596 DC-DC module of widespread use plugged into the IDT PROTO SHIELD PLUS



Figure 110 – Schematic of the External power supply section of the IOT PROTO SHIELD PLUS

## 35.1. Battery Shield

Alternatively, to the LM2596 module, you can use a typical battery shield module of widespread use (see Figure 111).

The battery shield module is used to boost the 3.7Vdc of a one element Lipo battery, up to 5V in order to supply it to the 5V pin of the ESP32 or Vin pin of the NANO boards (see Figure 114).

You connect the Lipo battery to the battery shield module by means of its JST connector.

To plug the battery shield module, you need two 8 pins header (not included in the IDT PROTO SHIELD PLUS) and solder them to the pads showed in Figure 112.

Then you can plug the battery shield module according to Figure 113.



Figure 111 – Typical battery shield of widespread use



Figure 112 - BRD5 socket of the IOT PROTO SHIELD PLUS



Figure 113 – Typical battery shield of widespread plugged into BRD5 socket of the IDT PROTO SHIELD PLUS

Battery_S	BRD5 Shield_8266	_mini	
×1 ×2		16 15	
×3 ×4		14 13 12	
×6 ×6 ×7 8	GND	11 10 9	-DGND
×	50		Vin

Figure 114 – Schematic of the battery shield section of the IOT PROTO SHIELD PLUS

IoT Proto Shield Plus User Manual Rev.1.07

## 36. The IoT Proto Shield Plus schematic design



*IoTPSP\_User\_Manual\_EN\_r1.0.docx* 

# www.GtronicsShop.com www.Gtronics.NET



Rev.1.07



IoTPSP\_User\_Manual\_EN\_r1.0.docx



IoTPSP\_User\_Manual\_EN\_r1.0.docx

105/106



IoTPSP\_User\_Manual\_EN\_r1.0.docx

106/106