# IOT PROTO SHIELD PLUS (VER. 1.0) MANUALE UTENTE



#### INDICE

| 1.    | Introduzione e specifiche tecniche   | . 7 |
|-------|--|-----|
| 2.    | Preparazione della la Iot Proto Shield Plus  | . 9 |
| 3.    | Layout   | 11  |
| 4.    | Importanti considerazioni sul voltaggio 5V e 3V3 prima di iniziare a lavorare con la IoT Proto |     |
| SHIE  | LD PLUS  | 14  |
| 4.1.  | Impostare Vbrd a 5V  | 14  |
| 5.    | Utilizzare la IDT PROTO SHIELD PLUS con schede Arduino NANO                                    | 16  |
| 6.    | Usare la IOT PROTO SHIELD PLUS con schede ESP32 MINI   | 17  |
| 7.    | Usare la IDT PROTO SHIELD PLUS con schede ESP8266 MINI   | 19  |
| 8.    | Connessioni di default sulla IDT PROTO SHIELD PLUS   | 20  |
| 9.    | Usare la IoT Proto Shield Plus con schede ESP32 o ESP8266                                      | 22  |
| 9.1.  | L'Adattatore UNO to Proto Shield Plus  | 22  |
| 9.2.  | Schede ESP32 o ESP8266 con fattore di forma differente   | 23  |
| 10.   | LEDs   | 24  |
| 10.1. | Connessioni di Default   | 25  |
| 10.2. | Sketch di esempio  | 25  |
| 10.3. | Usare i LED con pin GPIO differenti  | 25  |
| 11.   | РОТ  | 26  |
| 11.1. | Connessioni di default   | 27  |
| 11.2. | Sketch di esempio  | 27  |
| 11.3. | Usare il POTenziometro con pin GPIO differente   | 27  |
| 12.   | ENCODER  | 28  |
| 12.1. | Connessioni di default   | 29  |
| 12.2. | Connessioni consigliate per l'utilizzo dell'ENCODER con schede ESP8266                         | 29  |
| 12.3. | Sketch di esempio  | 30  |
| 12.4. | Utilizzare l'ENCODER con GPIO pin differenti   | 30  |
| 13.   | TOUCH  | 31  |
| 13.1. | Conessioni di default  | 32  |
| 13.2. | Connessioni consigliate per l'utilizzo del TOUCH con schede ESP8266                            | 32  |
| 13.3. | Sketch di esempio  | 32  |
| 13.4. | Utilizzare il TOUCH con GPIO pin differenti  | 33  |

| 14.   | I <sup>2</sup> C sulla IoT Proto Shield Plus                           | . 34 |  |  |  |  |  |  |
|-------|--|------|--|--|--|--|--|--|
| 14.1. | Connessioni dei default  |      |  |  |  |  |  |  |
| 14.2. | Scansionare il bus l <sup>2</sup> C per trovare i dispositivi connessi |      |  |  |  |  |  |  |
| 14.3. | Utilizzare il bus I <sup>2</sup> C con pin GPIO differenti             |      |  |  |  |  |  |  |
| 15.   | DISPLAY OLED SSD1306 128x64 (I <sup>2</sup> C)                         | . 36 |  |  |  |  |  |  |
| 15.1. | Connessioni di default   | . 37 |  |  |  |  |  |  |
| 15.2. | Sketch di esempio  | . 37 |  |  |  |  |  |  |
| 15.2  | 2.1. IoTPSP_SSD1306_DISPLAY_128x64_NANO                                | . 37 |  |  |  |  |  |  |
| 15.2  | 2.2. IoTPSP_SSD1306_DISPLAY_128x64_ESP32                               | . 37 |  |  |  |  |  |  |
| 15.2  | 2.3. IoTPSP_SSD1306_DISPLAY_128x64_ESP8266                             | . 38 |  |  |  |  |  |  |
| 15.3. | Utilizzare il display OLED SSD1306 128x64 con pin GPIO differenti      | . 38 |  |  |  |  |  |  |
| 16.   | DISPLAY OLED SSD1306 128x32 (I <sup>2</sup> C)                         | . 39 |  |  |  |  |  |  |
| 16.1. | Connessioni di default   | . 40 |  |  |  |  |  |  |
| 16.2. | Sketch di esempio  | . 40 |  |  |  |  |  |  |
| 16.2  | 2.1. IoTPSP_SSD1306_DISPLAY_128x32_NANO                                | . 40 |  |  |  |  |  |  |
| 16.2  | 2.2. IoTPSP_SSD1306_DISPLAY_128x32_ESP32                               | . 40 |  |  |  |  |  |  |
| 16.2  | 2.3. IoTPSP_SSD1306_DISPLAY_128x32_ESP8266                             | . 40 |  |  |  |  |  |  |
| 16.1. | Utilizzare il display OLED SSD1306 128x32 con pin GPIO differenti      | . 40 |  |  |  |  |  |  |
| 17.   | LCD 1602 (Liquid Crystal Display)                                      | . 41 |  |  |  |  |  |  |
| 17.1. | 17.1. Connessioni di default   |      |  |  |  |  |  |  |
| 17.2. | Sample Sketches  | . 42 |  |  |  |  |  |  |
| 17.3. | Cambiare l'indirizzo I <sup>2</sup> C del PCF8574                      | . 43 |  |  |  |  |  |  |
| 17.4. | Usare il PCF8574 con pin GPIO differenti                               | . 43 |  |  |  |  |  |  |
| 18.   | PCF8574 I2C I/O expander   | . 44 |  |  |  |  |  |  |
| 18.1. | Sketch di esempio  | . 45 |  |  |  |  |  |  |
| 18.1  | .1. IoTPSP_PCF8574_GET_ADDRESS   | . 45 |  |  |  |  |  |  |
| 19.   | TM1637 Digital display   | . 47 |  |  |  |  |  |  |
| 19.1. | Connessioni di default   | . 48 |  |  |  |  |  |  |
| 19.2. | Sample Sketches  | . 48 |  |  |  |  |  |  |
| 19.3. | Usare il display TM1637 con pin GPIO differenti                        | . 49 |  |  |  |  |  |  |
| 20.   | RELAY  | . 50 |  |  |  |  |  |  |
| 20.1. | Connessioni di default   | . 51 |  |  |  |  |  |  |

| Connessioni suggerite per lavorare con schede ESP8266                   |  |  |  |  |  |  |  |
|---|--|--|--|--|--|--|--|
| . Sketch di esempio   |  |  |  |  |  |  |  |
| Utilizzare moduli RELÈ con pin GPIO differenti                          |  |  |  |  |  |  |  |
| DHTxx (sensore di umidità e temperatura)53                              |  |  |  |  |  |  |  |
| 21.1. Connessioni di default  |  |  |  |  |  |  |  |
| 21.2. Connessioni suggerite per lavorare con schede ESP8266             |  |  |  |  |  |  |  |
| 21.3. Sketch di esempio   |  |  |  |  |  |  |  |
| 21.3.1. IoTPSP_DHT_NANO   |  |  |  |  |  |  |  |
| 21.3.2. IoTPSP_DHT_ESP32  |  |  |  |  |  |  |  |
| 21.3.3. IoTPSP_DHT_ESP8266  |  |  |  |  |  |  |  |
| 21.4. Utilizzare moduli e sensori DTHxx con pin GPIO differenti         |  |  |  |  |  |  |  |
| 22. RTC (Real Time Clock)   |  |  |  |  |  |  |  |
| 22.1. Connessioni di default  |  |  |  |  |  |  |  |
| 22.2. Sketch di esempio   |  |  |  |  |  |  |  |
| 22.3. Usare I moduli RTC con pin GPIO differenti                        |  |  |  |  |  |  |  |
| 23. Bus di comunicazione SPI sulla IDT PROTO SHIELD PLUS                |  |  |  |  |  |  |  |
| 23.1. Connessioni di default  |  |  |  |  |  |  |  |
| 24. MODULO SD CARD  |  |  |  |  |  |  |  |
| 24.1. Connessioni di default  |  |  |  |  |  |  |  |
| 24.2. Sketch di esempio   |  |  |  |  |  |  |  |
| .4.2.1. Testare la SD CARD  |  |  |  |  |  |  |  |
| 24.2.2. Leggere e scrivere su una scheda SD CARD 64                     |  |  |  |  |  |  |  |
| 25. LDR/xTC   |  |  |  |  |  |  |  |
| 25.1. Connessioni di default  |  |  |  |  |  |  |  |
| 25.2. Connessioni suggerite per lavorare con LDR/xTC con schede ESP8266 |  |  |  |  |  |  |  |
| 25.3. Sketch di esempio   |  |  |  |  |  |  |  |
| 25.4. Utilizzare sensori LDR/xTC con un pin GPIO differente             |  |  |  |  |  |  |  |
| Bus seriale aggiuntivo della Iot Proto Shield Plus69                    |  |  |  |  |  |  |  |
| 26.1. Connessioni di default  |  |  |  |  |  |  |  |
| 27. DFPlayer Mini MP3 Player70  |  |  |  |  |  |  |  |
| 27.1. Connessioni di default  |  |  |  |  |  |  |  |
| 27.2. Sketch di esempio   |  |  |  |  |  |  |  |
| 27.2.1. IOTPSP_MP3_NANO72   |  |  |  |  |  |  |  |

| 27.2   | .2.   | IoTPSP_MP3_NANO_EVERY_33  | 72   |
|--|---|---|--|
| 27.2   | .3.   | IoTPSP_MP3_ESP32  | 72   |
| 27.2   | .4.   | IoTPSP_MP3_ESP8266  | 73   |
| 27.3.  | Utilizzare  | e moduli DFPlayer Mini MP3 con pin GPIO differenti  | 73   |
| 27.3   | .1.   | IoTPSP_MP3_NANO_SoftwareSerial  | 73   |
| 28.  | SPEAKER   |   | 75   |
| 28.1.  | Conness   | ioni di default   | 75   |
| 28.2.  | Sketch d  | i esempio   | 76   |
| 28.2   | .1.   | IoTPSP_SPEAKER_NANO   | 76   |
| 28.2   | .2.   | IoTPSP_SPEAKER_ESP32  | 76   |
| 28.2   | .3.   | IoTPSP_SPEAKER_ESP8266  | 76   |
| 29.  | MIC   |   | 77   |
| 29.1.  | Modulo  | microfonico MAX9814   | 77   |
| 29.1.1   | . Impost  | are il guadagno (gain) del modulo   | 79   |
| 29.1.2   | 2. Impost   | are la risoluzione del modulo   | 79   |
| 29.2.  | Modulo  | microfonico MAX4466   | 79   |
|  |   |   |  |
| 29.3.  | Conness   | ioni di default   | 80   |
| 29.3.<br>29.4.   | Conness<br>Conness  | ioni di default<br>ioni suggerite per lavorare con moduli MIC e schede ESP8266  | 80<br>80   |
| 29.3.<br>29.4.<br>29.5.  | Conness<br>Conness<br>Sketch d  | ioni di default<br>ioni suggerite per lavorare con moduli MIC e schede ESP8266<br>i esempio   | 80<br>80<br>80   |
| 29.3.<br>29.4.<br>29.5.<br>29.6.   | Connessi<br>Connessi<br>Sketch d<br>Utilizzare  | ioni di default<br>ioni suggerite per lavorare con moduli MIC e schede ESP8266<br>i esempio<br>e i moduli MIC con pin GPIO differenti   | 80<br>80<br>80<br>80   |
| 29.3.<br>29.4.<br>29.5.<br>29.6.<br>30.  | Connessi<br>Connessi<br>Sketch d<br>Utilizzare<br>SERVO   | ioni di default<br>ioni suggerite per lavorare con moduli MIC e schede ESP8266<br>i esempio<br>e i moduli MIC con pin GPIO differenti   | 80<br>80<br>80<br>80<br>82   |
| <ol> <li>29.3.</li> <li>29.4.</li> <li>29.5.</li> <li>29.6.</li> <li>30.</li> <li>30.1.</li> </ol>   | Connessi<br>Connessi<br>Sketch d<br>Utilizzare<br>SERVO<br>Connessi   | ioni di default<br>ioni suggerite per lavorare con moduli MIC e schede ESP8266<br>i esempio<br>e i moduli MIC con pin GPIO differenti<br>ioni default   | 80<br>80<br>80<br>80<br>82<br>82   |
| <ol> <li>29.3.</li> <li>29.4.</li> <li>29.5.</li> <li>29.6.</li> <li>30.</li> <li>30.1.</li> <li>30.2.</li> </ol>  | Connessi<br>Sketch d<br>Utilizzare<br>SERVO<br>Connessi<br>Connessi   | ioni di default<br>ioni suggerite per lavorare con moduli MIC e schede ESP8266<br>i esempio<br>e i moduli MIC con pin GPIO differenti<br>ioni default<br>ioni suggerite per lavorare con schede ESP8266   | 80<br>80<br>80<br>82<br>82<br>82<br>83   |
| <ol> <li>29.3.</li> <li>29.4.</li> <li>29.5.</li> <li>29.6.</li> <li>30.</li> <li>30.1.</li> <li>30.2.</li> <li>30.3.</li> </ol>   | Connessi<br>Sketch d<br>Utilizzare<br>SERVO<br>Connessi<br>Connessi<br>Sketch d   | ioni di default<br>ioni suggerite per lavorare con moduli MIC e schede ESP8266<br>i esempio<br>e i moduli MIC con pin GPIO differenti<br>ioni default<br>ioni suggerite per lavorare con schede ESP8266<br>i esempio  | 80<br>80<br>80<br>82<br>82<br>83<br>83   |
| <ol> <li>29.3.</li> <li>29.4.</li> <li>29.5.</li> <li>29.6.</li> <li>30.</li> <li>30.1.</li> <li>30.2.</li> <li>30.3.</li> <li>30.3</li> </ol>   | Connessi<br>Sketch d<br>Utilizzare<br>SERVO<br>Connessi<br>Connessi<br>Sketch d<br>. 1 .  | ioni di defaultioni suggerite per lavorare con moduli MIC e schede ESP8266i<br>i esempioe i moduli MIC con pin GPIO differenti<br>ioni defaultioni suggerite per lavorare con schede ESP8266i<br>i esempio  | 80<br>80<br>80<br>82<br>82<br>83<br>83<br>83   |
| <ol> <li>29.3.</li> <li>29.4.</li> <li>29.5.</li> <li>29.6.</li> <li>30.</li> <li>30.1.</li> <li>30.2.</li> <li>30.3.</li> <li>30.3</li> <li>30.3</li> <li>30.3</li> </ol>   | Connessi<br>Sketch d<br>Utilizzare<br>SERVO<br>Connessi<br>Sketch d<br>.1.<br>.2.   | ioni di default<br>ioni suggerite per lavorare con moduli MIC e schede ESP8266<br>i esempio<br>e i moduli MIC con pin GPIO differenti<br>ioni default<br>ioni suggerite per lavorare con schede ESP8266<br>i esempio<br>I oTPSP_SERVO_NANO<br>I oTPSP_SERVO_ESP32   | <ul> <li>80</li> <li>80</li> <li>80</li> <li>82</li> <li>82</li> <li>83</li> <li>83</li> <li>83</li> <li>84</li> </ul>                                     |
| <ol> <li>29.3.</li> <li>29.4.</li> <li>29.5.</li> <li>29.6.</li> <li>30.</li> <li>30.1.</li> <li>30.2.</li> <li>30.3.</li> <li>30.3</li> <li>30.3</li> <li>30.3</li> <li>30.3</li> <li>30.3</li> </ol>                               | Connessi<br>Sketch d<br>Utilizzare<br>SERVO<br>Connessi<br>Sketch d<br>.1.<br>.2.<br>.3.  | ioni di default<br>ioni suggerite per lavorare con moduli MIC e schede ESP8266<br>i esempio<br>e i moduli MIC con pin GPIO differenti<br>ioni default<br>ioni suggerite per lavorare con schede ESP8266<br>i esempio<br>IOTPSP_SERVO_NANO<br>IOTPSP_SERVO_ESP32<br>IOTPSP_SERVO_ESP32   | <ul> <li>80</li> <li>80</li> <li>80</li> <li>82</li> <li>82</li> <li>83</li> <li>83</li> <li>84</li> <li>85</li> </ul>                                     |
| <ul> <li>29.3.</li> <li>29.4.</li> <li>29.5.</li> <li>29.6.</li> <li>30.</li> <li>30.1.</li> <li>30.2.</li> <li>30.3.</li> <li>30.3</li> <li>30.3</li> <li>30.3</li> <li>30.3</li> <li>30.3</li> <li>30.4.</li> </ul>                | Connessi<br>Sketch d<br>Utilizzare<br>SERVO<br>Connessi<br>Sketch d<br>.1.<br>.2.<br>.3.<br>Utilizzare                                    | ioni di default<br>ioni suggerite per lavorare con moduli MIC e schede ESP8266<br>i esempio   | <ul> <li>80</li> <li>80</li> <li>80</li> <li>82</li> <li>82</li> <li>83</li> <li>83</li> <li>84</li> <li>85</li> <li>85</li> </ul>                         |
| <ol> <li>29.3.</li> <li>29.4.</li> <li>29.5.</li> <li>29.6.</li> <li>30.</li> <li>30.1.</li> <li>30.2.</li> <li>30.3.</li> <li>30.3</li> <li>30.3</li> <li>30.3</li> <li>30.3</li> <li>30.4.</li> <li>30.5.</li> </ol>               | Connessi<br>Sketch d<br>Utilizzare<br>SERVO<br>Connessi<br>Sketch d<br>.1.<br>.2.<br>.3.<br>Utilizzare<br>Powering                        | ioni di default<br>ioni suggerite per lavorare con moduli MIC e schede ESP8266<br>i esempio<br>e i moduli MIC con pin GPIO differenti<br>ioni default<br>ioni suggerite per lavorare con schede ESP8266<br>i esempio<br>I oTPSP_SERVO_NANO<br>I oTPSP_SERVO_ESP32<br>I oTPSP_SERVO_ESP32<br>E un SERVO con pin GPIO differenti<br>g the SERVO with a different power supply | <ul> <li>80</li> <li>80</li> <li>80</li> <li>82</li> <li>82</li> <li>83</li> <li>83</li> <li>84</li> <li>85</li> <li>85</li> <li>85</li> </ul>             |
| <ol> <li>29.3.</li> <li>29.4.</li> <li>29.5.</li> <li>29.6.</li> <li>30.</li> <li>30.1.</li> <li>30.2.</li> <li>30.3.</li> <li>30.3</li> <li>30.3</li> <li>30.3</li> <li>30.3</li> <li>30.4.</li> <li>30.5.</li> <li>31.</li> </ol>  | Connessi<br>Sketch d<br>Utilizzare<br>SERVO<br>Connessi<br>Sketch d<br>.1.<br>.2.<br>.3.<br>Utilizzare<br>Powering<br>HC-SR04             | ioni di default<br>ioni suggerite per lavorare con moduli MIC e schede ESP8266<br>i esempio<br>e i moduli MIC con pin GPIO differenti<br>ioni default<br>ioni suggerite per lavorare con schede ESP8266<br>i esempio<br>IoTPSP_SERVO_NANO<br>IoTPSP_SERVO_ESP32<br>IoTPSP_SERVO_ESP32<br>IoTPSP_SERVO_ESP32 66<br>e un SERVO con pin GPIO differenti                        | <ul> <li>80</li> <li>80</li> <li>80</li> <li>82</li> <li>82</li> <li>83</li> <li>83</li> <li>84</li> <li>85</li> <li>85</li> <li>86</li> </ul>             |
| <ol> <li>29.3.</li> <li>29.4.</li> <li>29.5.</li> <li>29.6.</li> <li>30.</li> <li>30.1.</li> <li>30.2.</li> <li>30.3.</li> <li>30.3</li> <li>30.3</li> <li>30.3</li> <li>30.4.</li> <li>30.5.</li> <li>31.</li> <li>31.1.</li> </ol> | Connessi<br>Sketch d<br>Utilizzare<br>SERVO<br>Connessi<br>Sketch d<br>.1.<br>.2.<br>.3.<br>Utilizzare<br>Powering<br>HC-SR04<br>Connessi | ioni di default<br>ioni suggerite per lavorare con moduli MIC e schede ESP8266<br>i esempio<br>e i moduli MIC con pin GPIO differenti<br>ioni default<br>ioni suggerite per lavorare con schede ESP8266<br>i esempio  | <ul> <li>80</li> <li>80</li> <li>80</li> <li>82</li> <li>82</li> <li>83</li> <li>83</li> <li>84</li> <li>85</li> <li>85</li> <li>86</li> <li>87</li> </ul> |

| 31.3.  | Sketch di esempio   | 88  |  |  |  |  |
|--------|---|-----|--|--|--|--|
| 31.4.  | Utilizzare moduli HC-SR04 con pin GPIO differenti                           |     |  |  |  |  |
| 32.    | MOTORE PASSO-PASSO  |     |  |  |  |  |
| 32.1.  | Connessioni di default  | 90  |  |  |  |  |
| 32.2.  | Connessioni suggerite per lavorare con schede ESP8266                       | 91  |  |  |  |  |
| 32.3.  | Sketch di esempio   | 91  |  |  |  |  |
| 32.3   | .1. IoTPSP_STEPPER_MOTOR_NANO   | 91  |  |  |  |  |
| 32.3   | .2. IoTPSP_STEPPER_MOTOR_ESP32  | 92  |  |  |  |  |
| 32.3   | .3. IoTPSP_STEPPER_MOTOR_ESP8266  | 92  |  |  |  |  |
| 32.4.  | Scelta del modulo driver  | 93  |  |  |  |  |
| 32.4.1 | . Modulo DRV8825  | 93  |  |  |  |  |
| 32.4.2 | 2. Moduli TMC2xxx   | 94  |  |  |  |  |
| 32.4.3 | 8. Modulo DRV8834   | 95  |  |  |  |  |
| 32.4.4 | l. Modulo A4988   | 96  |  |  |  |  |
| 32.5.  | Scelta e connessione del motore passo-passo                                 | 98  |  |  |  |  |
| 32.6.  | Utilizzare motori passo-passo con pin GPIO differenti e funzioni aggiuntive | 98  |  |  |  |  |
| 33.    | LEVEL SHIFTER   | 99  |  |  |  |  |
| 33.1.  | Utilizzare il level shifter integrato con tensioni HV e LV differenti1      | .00 |  |  |  |  |
| 34.    | Modulo WiFi ESP01 1   | .02 |  |  |  |  |
| 35.    | ALIMENTAZIONE ESTERNA   | .04 |  |  |  |  |
| 35.1.  | Battery Shield  | .06 |  |  |  |  |
| 36.    | Schemi elettrici della Iot Proto Shield Plus1                               | .08 |  |  |  |  |

#### NOTE:

Alcune immagini contenute in questo manuale, potrebbero differire dalla revisione della scheda in vostro possesso (al momento della stesura la versione attuale è la rev. 1.0), ciò non pregiudica la funzionalità della scheda descritta in questo manuale.

## 1. Introduzione e specifiche tecniche

La scheda **IOT PROTO SHIELD PLUS** è progettata per progettare e prototipare facilmente con schede ESP32, ESP8266 e ARDUINO NANO.

Funziona con:

- Schede con fattore di forma Arduino NANO
  - o Arduino NANO 33 IoT
  - o Arduino NANO 33 BLE
  - o Arduino NANO RP2040 Connect
  - o Arduino NANO
  - o Arduino NANO EVERY
- Schede ESP32 con fattore di forma "MINI"
- Schede ESP8266 con fattore di forma "MINI"

Caratteristiche:

•

- Ampia area di breadboarding
  - o 2 aree di breadboarding da 740 punti di contatto ciascuna (totale 1480 punti)
  - Breakout headers per replicare ogni pin di I/O vicino all'area di breadboarding
- Breakout headers per replicare 3V3, 5V e GND vicino all'area di breadboarding
- Diverse tensioni di lavoro:
  - o 3V3
  - o 5V
  - o alimentazione esterna tramite modulo aggiuntivo LM2596 DC-DC di diffuso utilizzo
  - o alimentazione esterna tramite Battery Shield aggiuntivo di diffuso utilizzo
- Encoder rotativo integrato, con interruttore, pronto all'uso
- Interruttore Touch integrato, pronto per l'uso
- Due LED indicatori integrati pronti all'uso
- Potenziometro integrato con manopola, pronto all'uso
- Altoparlante integrato per riprodurre MP3 o toni
- Espansione I/O PCF8574 integrata, per pilotare LCD 16x2 in comunicazione I2C con LCD con soli 2 pin, pronta all'uso
- Area millefori di saldatura per ulteriori parti di prototipazione
- Quattro level shifter integrati, pronti per l'uso
- Pulsante Reset, per azzerare il programma e /o funzioni di programmazione
- Predisposizione Plug & Play per i seguenti moduli, sensori e attuatori di diffuso utilizzo:
  - Display LCD 16x2
  - o Display OLED SSD1306 128x64
  - Display OLED SSD1306 128x32
  - o Display a sette segmenti TM1637
  - RTC (orologio in tempo reale)
  - o Scheda SD
  - o Servo
  - o LDR (fotoresistenza dipendente dalla luce)
  - PTC o NTC (termistore a coefficiente di temperatura positivo o negativo)
  - o RELE'
  - o HC-SR04 (sensore di distanza ad ultrasuoni)
  - Modulo DTH11 / DHT22 o sensore DHT11 / DHT22 (sensore di umidità)

- Modulo MAX9814 o MAX4466 (modulo microfono)
- DFR0299 Mini (lettore di file musicali MP3)
- Dimensioni scheda in mm: 188x196x30 (HxLxA)

## 2. Preparazione della la IOT PROTO SHIELD PLUS

La scheda IoT PROTO SHIELD PLUS è composto da:

- 1x scheda IOT PROTO SHIELD PLUS
- 5x piedini in gomma adesivi
- 1x manopola del potenziometro

Inserire la manopola nel potenziometro (Figure 1) e incollare i piedini adesivi (Figure 2) nel caso non fossero già posizionati (dipende dal processo produttivo).



Figure 1 – Posizionamento della manopola del potenziometro

#### IoT Proto Shield Plus Manuale Utente



Figure 2 – Posizionamento dei 5 piedini adesivi

IOT PROTO SHIELD PLUS Manuale Utente

## 3. Layout



Figure 3 – Layout della IOT PROTO SHIELD PLUS

Si noti che la scheda IDT PROTO SHIELD PLUS funziona con una sola scheda controller alla volta (ovvero non è possibile collegare l'ESP32 ed un ARDUINO NANO contemporaneamente).

#### IOT PROTO SHIELD PLUS MANUALE UTENTE



Figure 4 – Lato superiore della IOT PROTO SHIELD PLUS



Figure 5 – Lato inferiore della IOT PROTO SHIELD PLUS

Rev.1.07.i

# 4. Importanti considerazioni sul voltaggio 5V e 3V3 prima di iniziare a lavorare con la IoT Proto Shield Plus

Le schede ESP32, ESP8266, NANO e NANO33 hanno una tensione di esercizio di 5V o 3,3V (d'ora in poi 3V3).

Gli ingressi di un dispositivo con tensione di esercizio 5V possono gestire un segnale con tensione 3V3 senza subire danneggiamenti, mentre gli ingressi di un dispositivo con tensione di esercizio 3V3 possono essere danneggiati se collegati a un segnale con tensione 5V.

Ecco perché sulla IDT PROTO SHIELD PLUS la tensione 3V3 è impostata come tensione predefinita per alimentazione di tutti i moduli Plug & Play che ad essa possono essere collegati.

La tensione operativa della IOT PROTO SHIELD PLUS viene identificata con Vbrd.



Figure 6 – Schema della connessione 3V3 o 5V a Vbrd

#### 4.1. Impostare Vbrd a 5V

Impostare Vbrd a 5V potreste danneggiare alcuni dispositivi con tensione di esercizio 3V3. Fare quindi molta attenzione e riferirsi ai data sheet delle schede e dei moduli in uso prima di impostare Vbrd a 5V.

Per impostare Vbrd a 5V: APRIRE (tagliare al centro) la connessione tra JP1 pad 1 e pad 2 e CHIUDERE (collegare con una piccola goccia di stagno) JP1 pad 2 a pad 3 (vedi Figure 9).



Figure 7 – JP1 della IOT PROTO SHIELD PLUS



Figure 8 – Impostare Vbrd a 3V3 (default)



Figure 9 – Impostare Vbrd a 5V

# 5. Utilizzare la IOT PROTO SHIELD PLUS con schede Arduino NANO

Inserire la scheda NANO nella IOT PROTO SHIELD PLUS, prestando attenzione all'allineamento e all'orientamento dei pin.

Collegare la scheda NANO tramite cavo USB al PC.

È importante considerare che le schede NANO e NANO EVERY funzionano a 5V, mentre le schede della famiglia NANO33 funzionano a 3V3.

Si consiglia di leggere le specifiche tecniche della scheda che si sta utilizzando per familiarizzare con essa.



Figure 10 – una scheda ARDUINO NANO (NANO33 IoT) inserita nella IoT PROTO SHIELD PLUS



Figure 11 – Schema connessioni dei pin ARDUINO NANO sulla IOT PROTO SHIELD PLUS

## 6. Usare la IDT PROTO SHIELD PLUS con schede ESP32 MINI

Inserire la scheda ESP32 MINI nella IoT PROTO SHIELD PLUS, prestando attenzione all'allineamento e all'orientamento dei pin.

Collegare la scheda ESP32 MINI tramite cavo USB al PC.

È importante considerare che le schede ESP32 funzionano a 3V3.

Si consiglia di leggere le specifiche tecniche della scheda che si sta utilizzando per familiarizzare con essa.



Figure 12 – La scheda ESP32 mini D1 inserita nella IoT PROTO SHIELD PLUS

|   |  |  |  |  | +5V .<br>↑                 | · Vin · ·  |      |   |  |
|---|--|--|--|--|----------------------------|--|------|---|--|
|   |  |  |  |  |                            | 3  |      |   |  |
|   |  |  | BRD7<br>mini_esp32   |  | -                          |  |      |   |  |
|   | CS<br>TDI<br>TCK<br>TMS<br>TDO<br>SCLK<br>MISO<br>SDA<br>SCL | GPI04         32           GPI05         14           GPI012         30           GPI013         18           GPI014         13           GPI015         37           GPI016         31           GPI017         29           GPI018         8           GPI019         10           GPI019         27           GPI022         25 | VCC_(USB)           GPI004         3V3           GPI005/CS         GND           GPI012/TDI         GND           GPI013/TCK         GND           GPI014/TMS         RST           GPI015/TD0         NC           GPI016         NC           GPI017         GPI018/SCLK           GPI019/MISO         GPI021/SDA           GPI022/SDA         GPI02/SCL | 35<br>16<br>33<br>22<br>1<br>2<br>1<br>5<br>3<br>3 | → +3V3<br>→ GND<br>• (R5T) | ♦ PWR_<br>· 5W1 ·<br>RESET<br><u>1</u> 4<br>2 3<br>· | FLAG | SCHEMATIC         ARDUINO         ESP32           (SERVO)         D5         GPI04           (SPL_CS)         D10         GPI05           (RELAY)         D8         GPI012           (DHT)         D9         GPI013           (HC_SR_T)         D6         GPI014           HC_SR_E         D7         GPI015           U_RX)         D0         GPI016           U_RX)         D1         GPI017           SPLSCLK         D13         GPI018           (LED_GRN)         4/6         GPI024 | J5<br>1 Pin3 1<br>2 Pin4 2<br>3 Pin5 3<br>4 Pin6 4<br>5 Pin7 5<br>6 Pin8 6<br>7 Pin9 7<br>8 Pin10 8<br>9 Pin11 9<br>10 Pin12 10<br>11 Pin13 11 |
| - | MOSI   | GPI023 12  | GPI022/SCL<br>GPI023/MOSI  |  |                            |  | GND  | A4 GPI021   | 12 Pin18 12  |
|   |  | GPI025 26<br>GPI026 6<br>GPI027 24   | GPI025 GPI000<br>GPI026 GPI001/TXD<br>GPI027 GPI002  | 34<br>21<br>36                                     | GPIO0<br>GPIO1<br>GPIO2    |  |      | LED_RED D11 GPI023<br>MOT_DIR D4 GPI025   | 13 Pin19 13<br>14 Pin20 14<br>15 Pin21 15  |
|   |  | GPI032 28<br>GPI033 9<br>GPI34 11  | GP1032 GP1003/RXD<br>GP1033  | 2 <u>3</u><br>40,                                  | GPIO3 <sub>e</sub>         |  |      | (MOT_STP) D2 GPI027<br>[ENC_A) D2 GPI027<br>[TOUCH] A0 GPI032   | 16 Pin22 16<br>17 Pin23 17   |
|   |  | GPI35 7<br>GPI36 4   | GPI34 GPI006/CLK<br>GPI35 GPI007/SD0<br>GPI36 GPI008/SD1   |  |                            |  |      | ENC_SW A1 GPI033<br>LDR_PTC A2 GPI34<br>A3 GPI35  | 19 Pin24 18<br>19 Pin25 19<br>20 Pin26 20  |
|   |  | <u>GPI39 5</u>   | GPI39 GPI009/SD2<br>GPI010/SD3<br>ESP32 GPI011/CMD   |  | -                          |  |      | POT A6 GPI36<br>MIC A7 GPI39  | 21 Pin27 21<br>22 Pin28 22   |

Figure 13 – Schema connessioni dei pin della scheda ESP32 sulla IoT PROTO SHIELD PLUS

## 7. Usare la IOT PROTO SHIELD PLUS con schede ESP8266 MINI

Il pinout della scheda ESP8266 MINI è una versione "ridotta" della scheda ESP32 (o il pinout della scheda ESP32 è una versione "estesa" della ESP8266....) pertanto la disposizione dei pin coincide in parte. Inserire la scheda ESP32 MINI nella IOT PROTO SHIELD PLUS, prestando attenzione all'allineamento e all'orientamento dei pin.

Collegare la scheda ESP32 MINI tramite cavo USB al PC.

È importante considerare che le schede ESP32 funzionano a 3V3.

Si consiglia di leggere le specifiche tecniche della scheda che si sta utilizzando per familiarizzare con essa.



Figure 14 – La scheda ESP8266 D1 inserita nella IOT PROTO SHIELD PLUS

00

# 8. Connessioni di default sulla IOT PROTO SHIELD PLUS

| В                   | DARDS PINOUT  |                 | IOT PROTO SHIELD PLUS |                        |                      |
|---------------------|---------------|-----------------|-----------------------|------------------------|----------------------|
| ESP 8266 ESP32 Mini |               | Arduino<br>NANO | Funzione<br>Primaria  | Funzione<br>Secondaria | JP di<br>riferimento |
|                     | GPIO00 / 100  |                 |                       |                        |                      |
| GPIO1 / TX          | GPIO01 / TXD  |                 |                       |                        |                      |
|                     | GPIO02 / IO2  |                 |                       |                        |                      |
| GPIO3 / RX          | GPIO03 / RXD  |                 |                       |                        |                      |
|                     | GPIO04 / 104  | 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 / IO25 | D4              | MOT_DIR               |                        | JP37                 |
| GPIO16 / D0         | GPIO26 / IO26 | D3              | MOT_STP               |                        | JP38                 |
|                     | GPIO27 / 1027 | D2              | ENC-A                 |                        | JP20                 |
|                     | GPIO32 / 1032 | A0              | TOUCH                 |                        | JP3                  |
|                     | GPIO33 / 1033 | A1              | ENC-SW                |                        | JP18                 |
|                     | GPIO34 / 1O34 | A2              | LDR / xTC             |                        | JP50                 |
|                     | GPIO35 / IO35 | A3              | ENC-B                 |                        | JP19                 |
| ADC0                | GPIO36 / SVP  | A6              | POT                   |                        | JP12                 |
|                     | GPIO39 / SVN  | A7              | MIC                   |                        | JP28                 |

HIGH DURANTE LA FASE DI BOOT

ERRORE DI BOOT SECOLLEGATO CON PULLDOWN

ERRORE DI BOOT SE COLLEGATO CON PULLUP

CONNESSO DI DEFAULT A DISPOSITIVI ON BOARD

LIBERO DI DEFAULT PER UTILIZZO GENERICO

#### HIGH DURANTE LA FASE DI BOOT

È da considerare che il pin sarà HIGH durante la fase di boot del dispositivo.

Es.

Se state usando una scheda ESP8266 e volete utilizzare il pin GPIO16/D0, dovete tenere in conto che tale pin sarà HIGH durante la fase di boot del modulo ESP8266.

#### ERRORE DI BOOT SECOLLEGATO CON PULLDOWN

È da considerare che, se collegate il pin ad una resistenza di PULL DOWN, il dispositivo non eseguirà il boot (quindi non funzionerà).

Es.

Se state usando una scheda ESP8266 e collegate il pin GPIOO/D3 ad una resistenza di pull down, il modulo ESP8266 non eseguirà il boot e quindi non funzionerà.

#### ERRORE DI BOOT SE COLLEGATO CON PULLUP

È da considerare che, se collegate il pin ad una resistenza di PULL UP, il dispositivo non eseguirà il boot (quindi non funzionerà).

Es.

Se state usando una scheda ESP8266 e collegate il pin GPIO12/DTDI ad una resistenza di pull up, il modulo ESP8266 non eseguirà il boot e quindi non funzionerà.

#### CONNESSO DI DEFAULT A DISPOSITIVI ON BOARD

È da considerare che il pin è collegato di default ad uno dei dispositivi integrati disponibili sulla IDT PROTO SHIELD PLUS.

Es.

Se state utilizzando una scheda NANO e volete utilizzare A6 per acquisire un segnale analogico, è da considerare che di default, il pin, è connesso al potenziometro integrato POT (RV2).

Per utilizzare A6 sarà quindi necessario disconnetterlo dal POT incidendo JP12 nel mezzo (si veda più avanti nel manuale la spiegazione per ogni dispositivo integrato).

#### LIBERO DI DEFAULT PER UTILIZZO GENERICO

Se il modulo o dispositivo relativo alla funzione primaria non è inserito nella IDT PROTO SHIELD PLUS, il pin è liberamente utilizzabile per i vostri scopi.

Es.

Se state utilizzando una scheda NANO e volete utilizzare A7 per acquisire un segnale analogico, se il modulo MIC non è inserito nel connettore, potete collegare tranquillamente il vostro segnale al pin A7.

#### CONCLUSIONI

- Se dovete utilizzare un modulo o dispositivo per cui è prevista la predisposizione plug and play, inserite il modulo o dispositivo nel connettore dedicato (si veda avanti nel manuale il dettaglio per ogni predisposizione) ed il dispositivo sarà già pronto all'uso.
- Se dovete utilizzare dispositivi o moduli per cui non è prevista la predisposizione, inseritelo nell'area millefori e collegatelo preferibilmente ad uno dei pin contrassegnati come "LIBERO DI DEFAULT PER UTILIZZO GENERICO", considerandone l'eventuale comportamento durante la fase di boot.
- Se il vostro progetto richiede necessariamente l'utilizzo di uno dei pin contrassegnati come "CONNESSO DI DEFAULT A DISPOSITIVI ON BOARD", disconnettetelo dal dispositivo integrato ed utilizzatelo secondo i vostri scopi facendo sempre le dovute considerazioni in merito all'eventuale comportamento durante la fase di boot.

## 9. Usare la IoT Proto Shield Plus con schede ESP32 o ESP8266

Per utilizzare la IoT PROTO SHIELD PLUS con schede con un fattore di forma differente da NANO (vedi §5) oppure ESP32 D1 MINI (vedi §6) oppure ESP8266 D1 MINI (vedi §7), è necessario utilizzare degli adattatori.

## 9.1. L'Adattatore UNO to Proto Shield Plus

Per utilizzare la IoT PROTO SHIELD PLUS con schede con fattore di forma di tipo UNO (es. Arduino UNO, Leonardo, Zero, Yun, ecc.), è possibile utilizzare l'Adattatore UNO to IoT Proto Shield Plus (https://www.gtronicsshop.com/it/proto-shield-plus/37-uno-to-iot-proto-shield-plus-adapter.html).

L'adattatore funziona con qualsiasi scheda Arduino UNO compatibile in termini di dimensioni, pin-out e specifiche elettriche.

Tuttavia, prima di utilizzare l'**Adattatore UNO to IoT Proto Shield Plus**, si raccomanda di prendere visione del manuale utente.



Figure 15 – L'adattatore UNO to IoT Proto Shield Plus

Per utilizzare la IOT PROTO SHIELD PLUS con schede con fattore di forma differente (come ad esempio quelle mostrate in Figure 16) è necessario utilizzare degli adattatori.

Nel caso, potete richiedere se un nuovo adattatore è disponibile oppure potete voi stessi progettare il vostro adattatore secondo le vostre esigenze prendendo come esempio l'adattatore mostrato al paragrafo precedente.



Figure 16 – Esempi di schede ESP32 e ESP8266 con fattore di forma differente dal MINI

IoT Proto Shield Plus Manuale Utente

## 10. LEDs

Utilizzare i due LED della IOT PROTO SHIELD PLUS.





Figure 17 – Sezione LED della IOT PROTO SHIELD PLUS



Figure 18 – Schema connessioni dei della sezione LED della IOT PROTO SHIELD PLUS

#### 10.1. Connessioni di Default

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

#### 10.2. Sketch di esempio

Posizione: IoTPSP\_Sample\_Sketches\IoTPSP\_LEDs

A seconda della scheda che si sta utilizzando, gli sketch fanno lampeggiare alternativamente il LED verde (GRN) ed il rosso (RED).

Usare lo sketch IoTPSP\_LEDs\_NANO per testare i LED con schede ARDUINO NANO. Usare lo sketch IoTPSP\_LEDs\_ESP32 per testare i LED con schede ESP32. Usare lo sketch IoTPSP\_LEDs\_ESP8266 per testare i LED con schede ESP8266.

## 10.3. Usare i LED con pin GPIO differenti

Come mostrato in Figure 18, i LED GRN e RED sono collegati di default ai pin GPIO attraverso JP11 e JP9. Per disconnettere il LED GRN dal pin GPIO di default, APRIRE JP11 (incidere nel centro).

Per disconnettere il LED RED dal pin GPIO di default, APRIRE JP9 (incidere nel centro).

Saldare un header su J6 e, tramite dei cavetti volanti, collegare i LED GRN e RED ad altri pin GPIO secondo le proprie esigenze.

Verificare le connessioni di default della IOT PROTO SHIELD PLUS (vedere §8) per evitare conflitti con altre connessioni predefinite.

IOT PROTO SHIELD PLUS Manuale Utente

## 11. POT

Utilizzare il POTenziometro della IOT PROTO SHIELD PLUS.





Figure 19 - Sezione POT della IOT PROTO SHIELD PLUS



Figure 20 – Diagramma connessioni della sezione POT della IoT PROTO SHIELD PLUS

#### 11.1. Connessioni di default

|     | ESP32  | ARDUINO | ESP8266 |
|-----|--------|---------|---------|
| РОТ | GPIO36 | A6      | A0      |

## 11.2. Sketch di esempio

Posizione: IoTPSP Sample Sketches\IoTPSP POT

A seconda della scheda che si sta utilizzando, gli sketch leggono il valore analogico del POTenziometro e stampano il valore letto sul serial monitor.

Usare lo sketch IoTPSP\_POT\_NANO per testare il POTenziometro con schede ARDUINO NANO. Usare lo sketch IoTPSP\_POT\_ESP32 per testare il POTenziometro con schede ESP32. Usare lo sketch IoTPSP\_POT\_ESP8266 per testare il POTenziometro con schede ESP8266.

## 11.3. Usare il POTenziometro con pin GPIO differente

Come mostrato in Figure 20, il POTenziometro è collegato di default al pin GPIO attraverso JP12. Per disconnettere il POTenziometro dal pin GPIO di default, aprire JP12 (incidere nel mezzo). Saldare un header su J7 e, tramite dei cavetti volanti, collegare il POTenziometro ad un altro pin GPIO secondo le proprie esigenze.

Verificare le connessioni default della IOT PROTO SHIELD PLUS (vedere §8) per evitare conflitti con altre connessioni predefinite.

IOT PROTO SHIELD PLUS Manuale Utente

## **12. ENCODER**

Utilizzare l' ENCODER della IOT PROTO SHIELD PLUS.





Figure 21 – Sezione ENCODER della IOT PROTO SHIELD PLUS



Figure 22 – Diagramma collegamenti della sezione ENCODER della IoT PROTO SHIELD PLUS

#### 12.1. Connessioni di default

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

*NC* = Non Connesso

Si noti che non ci sono connessioni predefinite per schede ESP8266. Per utilizzare l'ENCODER con schede ESP8266, è necessario eseguire i collegamenti con dei cavetti volanti.

#### 12.2. Connessioni consigliate per l'utilizzo dell'ENCODER con schede ESP8266

Gli sketch di esempio funzionano con le seguenti connessioni: J3 pin GPIO14 (D5) dell'ESP8622 connesso al pin GPIO33 dell'ESP32 (ENC\_SW) J3 pin GPIO5 (D1) dell'ESP8622 connesso al pin GPIO27 dell'ESP32 (ENC\_A) J3 pin GPIO4 (D2) ESP8622 connesso al pin GPIO35 dell'ESP32 (ENC\_B)



Figure 23 – Connessioni consigliate per l'utilizzo dell'ENCODER con schede ESP8266

#### 12.3. Sketch di esempio

Posizione: IoTPSP\_Sample\_Sketches\IoTPSP\_ENCODER

Usare lo sketch IoTPSP\_ENCODER\_Simple\_NANO per testare l'ENCODER con schede ARDUINO NANO. Usare lo sketch IoTPSP\_ENCODER\_Simple\_ESP32 per testare l'ENCODER con schede ESP32. Usare lo sketch IoTPSP\_ENCODER\_Simple\_ESP8266 per testare l'ENCODER con schede ESP8266.

Aprire il serial monitor e ruotare la manopola dell'ENCODER, si vedrà la posizione incrementale aumentare o diminuire a seconda del senso di rotazione, premere la manopola per accendere il LED GRN.

Utilizzando gli interrupt è possibile implementare in modo più robusto la gestione dei segnali A e B dell'ENCODER.

Usare lo sketch IoTPSP\_ENCODER\_Interrupt\_NANO per testare l'ENCODER con gestione dei segnali A e B con interrupt con schede ARDUINO NANO.

Usare lo sketch IoTPSP\_ENCODER\_ Interrupt \_ESP32 per testare l'ENCODER con gestione dei segnali A e B con interrupt con schede ESP32.

Usare lo sketch IoTPSP\_ENCODER\_ Interrupt \_ESP8266 per testare l'ENCODER con gestione dei segnali A e B con interrupt con schede ESP8266.

#### 12.4. Utilizzare l'ENCODER con GPIO pin differenti

Come mostrato in Figure 22, l'ENCODER è collegato ai pin GPIO di default attraverso JP18, JP19 e JP20. Per disconnettere l'ENOCDER dai pin GPIO di default, aprire JP18, JP19 e JP20 (incidere nel centro). Saldare un header in posizione J9 e, tramite dei cavetti volanti, collegare l'ENCODER ad altri pin GPIO secondo le proprie esigenze.

Verificare le connessioni default della IOT PROTO SHIELD PLUS (vedere §8) per evitare conflitti con altre connessioni predefinite.

IOT PROTO SHIELD PLUS MANUALE UTENTE

## **13. TOUCH**

Il sensore TOUCH della IDT PROTO SHIELD PLUS utilizza il chip TTP223E, lo stesso utilizzato largamente nelle diffusissime schedine touch.





Figure 24 – Sezione TOUCH della IDT PROTO SHIELD PLUS



Figure 25 – Digramma connessioni della sezione TOUCH della IDT PROTO SHIELD PLUS

#### 13.1. Conessioni di default

|       | ESP32  | ARDUINO | ESP8266 |  |  |  |
|-------|--------|---------|---------|--|--|--|
| TOUCH | GPIO32 | A0      | NC      |  |  |  |
|       |        |         |         |  |  |  |

*NC* = Non Connesso

Si noti che non ci sono connessioni predefinite per schede ESP8266. Per utilizzare il TOUCH con schede ESP8266, è necessario eseguire i collegamenti con dei cavetti volanti.

#### 13.2. Connessioni consigliate per l'utilizzo del TOUCH con schede ESP8266

Gli sketch di esempio funzionano con le seguenti connessioni: J3 pin GPIO14 (D5) dell'ESP8622 connesso al pin GPIO32 dell'ESP32 (TOUCH)



Figure 26 - Connessioni consigliate per l'utilizzo dell'ENCODER con schede ESP8266

#### 13.3. Sketch di esempio

Posizione: IoTPSP Sample Sketches\IoTPSP TOUCH

Usare lo sketch IoTPSP\_TOUCH\_NANO per testare il TOUCH con schede ARDUINO NANO. Usare lo sketch IoTPSP\_TOUCH\_ESP32 per testare il TOUCH con schede ESP32. Usare lo sketch IoTPSP\_TOUCH\_ESP8266 per testare il TOUCH con schede ESP8266.

Toccando il touch sui cerchi concentrici il LED GRN si accenderà.

La scheda ESP32 ha anche un sensore TOUCH interno, è pertanto possibile escludere il chip TTP223E e lavorare direttamente con il sensore interno collegato al GPIO32 (TOUCH9).

Per lavorare con il sensore interno della scheda ESP32, è necessario APRIRE JP22 (incidere nel centro), CHIUDERE i pad 3 e 2 di JP25 (connettere i due ad con una piccola goccia di stagno) e APRIRE pad 1 e 2 (incidere tra i due pad).

Usare lo sketch IoTPSP\_TOUCH\_ESP32\_INTERNAL per testare il TOUCH interno con schede ESP32. Utilizzare il serial monitor per visualizzare il valore letto dal sensore interno.

# 13.4. Utilizzare il TOUCH con GPIO pin differenti

Come mostrato in Figure 25, il segnale del sensore TOUCH è collegato al pin GPIO di default attraverso JP3. Per disconnettere il TOUCH dal pin GPIO di default, APRIRE JP3 (incidere nel centro).

Saldare un header in posizione J43 e, tramite dei cavetti volanti, collegare il TOUCH ad un altro pin GPIO secondo le proprie esigenze.

Verificare le connessioni default della IOT PROTO SHIELD PLUS (vedere §8) per evitare conflitti con altre connessioni predefinite.

## 14. I<sup>2</sup>C sulla IOT PROTO SHIELD PLUS

I<sup>2</sup>C (Inter-Integrated Circuit, eye-squared-C) è un protocollo di comunicazione seriale largamente usato per connettere differenti dispositivi elettronici (è possibile trovare maggiori informazioni relative al bus I<sup>2</sup>C su <u>Wikipedia</u> o altra documentazione on line).

Il protocollo I<sup>2</sup>C sulla I**DT PROTO SHIELD PLUS** è utilizzato per comunicare con dispositivi quali displays, RTC clock, ecc.



Figure 27 – Schema collegamento segnali del bus I<sup>2</sup>C sulla IOT PROTO SHIELD PLUB

#### 14.1. Connessioni dei default

I segnali del bus l<sup>2</sup>C sulla IoT PROTO SHIELD PLUS sono connessi di default ai seguenti pin:

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

#### 14.2. Scansionare il bus l<sup>2</sup>C per trovare i dispositivi connessi

Utilizzare lo sketch di esempio IoTPSP\_I2C\_scanner (posizione:

IOTPSP\_Sample\_Sketches\IoTPSP\_I2C\_scanner) per scansionare il bus l<sup>2</sup>C. Lo scanner mostra gli indirizzi dei dispositivi l<sup>2</sup>C presenti sul bus. Senza Nessun dispositivo aggiuntivo inserito (ad esempio display OLED, ecc.) lo scanner mostra l'indirizzo del chip PCF8574 presente sulla IoT PROTO SHIELD PLUS (tipicamente 0x20 o 0x38).

## 14.3. Utilizzare il bus l<sup>2</sup>C con pin GPIO differenti

La maggior parte delle librerie utilizzate per comunicare con dispositivi l<sup>2</sup>C, utilizzano i pin GPIO di default (vedi §14.1), quindi **è SCONSIGLIATO modificare le connessioni del bus l<sup>2</sup>C**.

Tuttavia, se proprio è necessario, di seguito è indicato come procedere.

Come mostrato in Figure 27, i segnali del bus I<sup>2</sup>C sono collegati ai pin GPIO di default attraverso JP6 e JP7.

Per disconnettere il bus dai pin GPIO di default, APRIRE JP6 e JP7 (incidere nel centro). Saldare un header in posizione J41 e, tramite dei cavetti volanti, collegare i segnali SDA e SCL ad altri pin GPIO secondo le proprie esigenze.

Verificare le connessioni default della IOT PROTO SHIELD PLUB (vedere §8) per evitare conflitti con altre connessioni predefinite.

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

La **IDT PROTO SHIELD PLUS** è predisposta per lavorare con i display SSD1306 OLED 128x64 a 4 pin (I<sup>2</sup>C) largamente diffusi sul mercato.

Per lavorare con un display SSD1306 OLED 128x64 a 4 pin (I<sup>2</sup>C), inserirlo nell'header BRD9 come mostrato in Figure 30.



Figure 28 – L'header BRD9 dedicato al DISPLAY OLED 128x64 della IDT PROTO SHIELD PLUS



Figure 29 – Schema collegamenti dell'header BRD9, dedicato al DISPLAY OLED 128x64 della IoT PROTO SHIELD PLUS

| LCD 16x | VDG:  | 20<br>25 | 2 - 2 | 3 3 3  | 88    | 0.5<br>0 | IPI<br>I<br>I<br>I<br>I<br>I<br>I<br>I<br>I<br>I<br>I<br>I<br>I<br>I<br>I<br>I<br>I<br>I<br>I |   |   | J Ó |
|---------|-------|----------|-------|--------|-------|----------|---|---|---|-----|
| 0000    | 00000 | 00000    |       | 000000 | 00000 | 00000    | SDA<br>SCK<br>Vcc (<br>GND<br>Brout 00  |   |   |     |
| 0000    | 00000 | 0000     |       |        | 00000 | 00000    |   | 0 |   | LO  |
|         | 3_    |          |       |        |       |          |   |   | - |     |

Figure 30 – Display I<sup>2</sup>C OLED SSD1306 128x64 inserito nell'header BRD9 della IOT PROTO SHIELD PLUS
Si veda §14.1 relativo ai segnali del bus l<sup>2</sup>C sulla IOT PROTO SHIELD PLUS.

## 15.2. Sketch di esempio

Posizione: IoTPSP Sample Sketches\IoTPSP SSD1306 DISPLAY 128x64

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

Usare lo sketch IoTPSP\_SSD1306\_DISPLAY\_128x64\_NANO, per testare il display OLED SSD1306 128x64 con schede NANO.

Lo sketch fa uso della libreria *Adafruit SSD1306*. Assicuratevi di averla installata prima di testare lo sketch.

|   | Auto Format<br>Archive Sketch   | Ctrl+T  |      |
|---|---|---|------|
| IoTPSP_SSD1   | Fix Encoding & Reload   |   |      |
| **************************************  | Manage Libraries  | Ctrl+Shift+I  |      |
| the sketch m  | Serial Monitor  | Ctrl+Shift+M  |      |
| Library Manager   | 10.000  |   |      |
| 100   | AMALES - SALEND   |   |      |
| ype AI<br>ACROBOTIC SSD13<br>5y ACROBOTIC V<br>Library for SSD13<br>More info   | V Top: All V 1300<br>06<br>arsion 1.0.1 INSTALLED<br>00-powered OLED 138x84 displayst Library for   | SSD1306-powered OLED 128x64 displays/   |      |
| ype  Al<br>ACROBOTIC SSD13<br>by ACROBOTIC V<br>Library for SSD13<br>More info  | v Topic All v 1306<br>06<br>arcion 3.0.1 INSTALLED<br>06-powared OLED 128x04 displayst Library for  | SSD1306-powered OLED 128x64 displays1   |      |
| ype [A]<br>ACROBOTIC SSD13<br>by ACROBOTIC W<br>Library for SSD13<br>More info<br>by Adabruit SSD1306<br>by Adabruit SSD1306<br>by Adabruit Versis<br>SSD1306 claid dri<br>and 128:32 displi<br>More info | Top: All 2335  66  sector 1.0.1 TNSTALLED  00-powered OLED 128x04 displayet Library for  on 2.4.3 TNSTALLED  wer library for monochrome 128x64 and 128x3  wys                         | SSD1306-powered OLED 128x64 displayor<br>displays SSD1306 eled driver library for monochrome 128            | 1254 |
| ype [A]<br>ACROBOTIC SSD13<br>by ACROBOTIC SSD13<br>by ACROBOTIC V<br>Library for SSD13<br>Mora info<br>SSD1306 al-d dri-<br>and 128-32 displ<br>Mora info<br>Select version v                            | v Topic All V 2335<br>66<br>arction 10.0.1 TINSTALLED<br>00-powered OLED 128x64 displayet Library for<br>an 2,4-3 TINSTALLED<br>are tilterary for monochromes 128x64 and 128x3<br>was | 5501306-powerad OLED 128x64 displays1<br>displays 9501306, aled driver library for monochrome 128<br>Ubdate | l=64 |

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

Usare lo sketch <code>IoTPSP\_SSD1306\_DISPLAY\_128x64\_ESP32</code>, per testare il display OLED SSD1306 128x64 con schede ESP32. Lo sketch fa uso della libreria *ESP8266 and ESP32 OLED driver for SSD1306 displays*.

Assicuratevi di averla installata prima di testare lo sketch.

| IoTPSP_SSD13            | 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  |     |
| CSP Generate QRCOUP for GDEH0213572 enink display, 5501300, 541100 oned displays 128 of pixer and others based on<br>Adafruit ST7ZXX ESP Generate QRCode for GDEH0213872 eink display, SSD1306, SH1106 oled displays 128*64 pixel and<br>others based on Adafruit ST77XX<br>More info  | ^   |
| ESP8266 and ESP32 OLED driver for SSD1306 displays   |     |
| by ThingPulse, Fabrice Weinberg Version 4.2.0 INSTALLED<br>I2C display driver for SSD1306 OLED displays connected to ESP8266, ESP32, Mbed-OS The following geometries are currently<br>supported: 128x64, 128x32, 64x48. The init sequence was inspired by Adafruit's library for the same display.<br>More info<br>Select version v Install |     |
| CCDD2CC QDarda   | -   |
| by Anun Panya<br>ESPP266 Generate QRCode for SSD1306 oled displays 128*64 pixel ESP8266 Generate QRCode version 7 for SSD1306 oled<br>displays 128*64 pixel<br>More info   |     |
|  | •   |
| d  | ose |

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

Usare lo sketch IoTPSP\_SSD1306\_DISPLAY\_128x64\_ESP8266 per testare il display OLED SSD1306 128x64 con schede ESP8266. Lo sketch fa uso della libreria *ESP8266 and ESP32 OLED driver for SSD1306 displays*.

Assicuratevi di averla installata prima di testare lo sketch. (vedi § 15.2.2).

### 15.3. Utilizzare il display OLED SSD1306 128x64 con pin GPIO differenti

Si veda § 14.3 relativo a come utilizzare il bus I<sup>2</sup>C con pin GPIO differenti sulla IOT PROTO SHIELD PLUS.

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

La **IDT PROTO SHIELD PLUS** è predisposta per lavorare con i display SSD1306 OLED 128x32 a 4 pin (I<sup>2</sup>C) largamente diffusi sul mercato.

Per lavorare con un display SSD1306 OLED 128x32 a 4 pin (I<sup>2</sup>C), inserirlo nell'header BRD10 come mostrato in Figure 33.



Figure 31 – L'header BRD10 dedicato al DISPLAY OLED 128x32 della IOT PROTO SHIELD PLUS



Figure 32 – Schema collegamenti dell'header BRD10, dedicato al DISPLAY OLED 128x32 della IDT PROTO SHIELD PLUS

| 16x      | 200 S | RS RS | <br>3 8 | 2 2 | 8 8                                     | 5            |       | JRD9 | π |
|----------|-------|-------|---------|-----|---|--------------|-------|------|---|
| 00000000 |       |       |         |     | 000000000000000000000000000000000000000 | an harren an | Leeba |      |   |
|          |       |       |         |     |   |              |       |      | 1 |

Figure 33 – Display I<sup>2</sup>C OLED SSD1306 128x32 inserito nell'header BRD10 della IOT PROTO SHIELD PLUS

Si veda §14.1 relativo ai segnali del bus l<sup>2</sup>C sulla IOT PROTO SHIELD PLUS.

## 16.2. Sketch di esempio

Posizione: IoTPSP Sample Sketches\IoTPSP SSD1306 DISPLAY 128x32

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

Usare lo sketch IoTPSP\_SSD1306\_DISPLAY\_128x32\_NANO, per testare il display OLED SSD1306 128x32 con schede NANO. Lo sketch fa uso della libreria *Adafruit SSD1306*. Assicuratevi di averla installata prima di testare lo sketch (vedi § 15.2.1).

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

Usare lo sketch <code>IoTPSP\_SSD1306\_DISPLAY\_128x32\_ESP32</code>, per testare il display OLED SSD1306 128x32 con schede ESP32. Lo sketch fa uso della libreria *ESP8266 and ESP32 OLED driver for SSD1306 displays*.

Assicuratevi di averla installata prima di testare lo sketch (vedi § 15.2.2).

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

Usare lo sketch IoTPSP\_SSD1306\_DISPLAY\_128x32\_ESP8266 per testare il display OLED SSD1306 128x32 con schede ESP8266. Lo sketch fa uso della libreria *ESP8266 and ESP32 OLED driver for SSD1306 displays*.

Assicuratevi di averla installata prima di testare lo sketch. (vedi § 15.2.2).

### 16.1. Utilizzare il display OLED SSD1306 128x32 con pin GPIO differenti

Si veda § 14.3 relativo a come utilizzare il bus I<sup>2</sup>C con pin GPIO differenti sulla IOT PROTO SHIELD PLUS.

# 17. LCD 1602 (Liquid Crystal Display)

La **IOT PROTO SHIELD PLUS** è predisposta per lavorare con display LCD 1602 a 16 pin largamente diffusi sul mercato.

Per lavorare con un display LCD 1602, è sufficiente collegare il modulo all'header LCD16x2 come mostrato nella Figure 35.

Utilizzare RV1 per regolare il contrasto dell'LCD.

Il display LCD 1602 è pilotato con bus I<sup>2</sup>C tramite il chip PCF8574(I<sup>2</sup>C I/O expander) integrato sulla IDT PROTO SHIELD PLUS, non è richiesto alcun collegamento aggiuntivo.

Le connessioni dei segnali l<sup>2</sup>C sono le stesse utilizzate nell'adattatore largamente diffuso sul mercato. Poiché le schede ESP32, ESP8266 e NANO33 funzionano a 3,3 V, per avere un contrasto sufficiente sul display, viene utilizzato un ICL7660 per garantire una tensione sufficiente al pin V0 del modulo LCD 1602. A seconda della disponibilità dei componenti, I/O expander può essere il PCF8574 o il PCF8574A. Entrambe le versioni funzionano allo stesso modo, l'unica differenza è l'indirizzo di base della comunicazione l<sup>2</sup>C.

Poiché AO, A1 e A2 del PCF8574 sono collegati a GND di default (tramite JP17, JP16 e JP15), l'indirizzo predefinito è impostato su 0x20 (PCF8574) o 0x38 (PCF8574A).

Usare gli sketch di esempio riportati di seguito (vedi § 17.2) per scoprire quale versione è montata sulla scheda IoT PROTO SHIELD PLUS in vostro possesso.



Figure 34 – Header LCD 16x2 della IOT PROTO SHIELD PLUS



Figure 35 – Un display LCD 1602 inserito nell'header 16 pin della IoT PROTO SHIELD PLUS



Figure 36 – Schema collegamenti della sezione LCD 16x2 della IDT PROTO SHIELO PLUS

Si veda §14.1 relativo ai segnali del bus l<sup>2</sup>C sulla IOT PROTO SHIELD PLUS.

## 17.2. Sample Sketches

**Posizione:** IoTPSP\_Sample\_Sketches\IoTPSP\_LCD16x2

Questi sketch di esempio fanno uso della libreria *LiquidCrystal\_I2C library*, la libreria è presente nel percorso IoTPSP\_Sample\_Sketches\IoTPSP\_LCD16x2.

Questi sketch di esempio usano le connessioni I<sup>2</sup>C mostrate al § 14.1.

Usare lo sketch IoTPSP\_LCD16x2\_NANO per testare il display LCD1602 con schede ARDUINO NANO. Usare lo sketch IoTPSP\_LCD16x2\_ESP32 per testare il display LCD1602 con schede ESP32. Usare lo sketch IoTPSP\_LCD16x2\_ESP8266 per testare il display LCD1602 con schede ESP8266.

## 17.3. Cambiare l'indirizzo l<sup>2</sup>C del PCF8574

Non è molto usuale, ma nel caso in cui si voglia/si debba cambiare l'indirizzo del chip 8574, bisogna APRIRE o CHIUDERE JP17, JP16 e JP15 in base alle proprie esigenze e al datasheet del PCF8574. Per APRIRE JP17, JP16 o JP15 incider al centro. Per CHIUDERE utilizzare una goccia di stagno.

## 17.4. Usare il PCF8574 con pin GPIO differenti

Si veda § 14.3 relativo a come utilizzare il bus l<sup>2</sup>C con pin GPIO differenti sulla IoT Proto Shield Plus.

# 18. PCF8574 I2C I/O expander

Nel caso non si utilizzi il display LCD 1602, è possibile utilizzare il PCF8574 come I/O expander. Tutti i pin I/O del PCF8574 sono disponibili sul lato destro della IDT PROTO SHIELD PLUS (vedere Figure 37).



Figure 37 – I pin del PCF8574 utilizzabili come general-purpose I/O

## 18.1. Sketch di esempio

**Posizione:** IoTPSP\_Sample\_Sketches\IoTPSP\_PCF8574

Questi sketch di esempio fanno uso delle connesioni I<sup>2</sup>C di default mostrate al §14.1. Questi sketch fanno uso della libreiria *PCF8574 library*, assicuratevi di averla installata rima di usare gli sketch.

| IoTPSP_SSD1306   | _DISPLAY_128x64_NANO   Arduir   | no 1.8.13  |   |
|--|---|--|---|
| ile Edit Sketch To   | ols 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 Manage   | r   |  | ×   |
| Гуре All   | V Topic All   | v <b>8574</b>  |   |
|  |   |  | Version 1.2.0 V                           |
| More info  | Stanler 1201  | o capanoci ampicinenti sinit iotate an   | u uyyıtı                                  |
| PCF8574 library<br>by Renzo Mischi<br>Arduino, ESP82<br>read wite digita<br><u>More info</u> | ianti<br>16 <b>6, smt32 and esp32 library f</b> r<br>Il values with only 2 wire. Very | or PCF8574 i2c digital expander for Ardu<br>simple to use and encoder support. | uino, esp32, SMT32 and ESP8266. Can       |
| by Ricardo Lima<br>Control the PCF   | ) Caratti, pu2clr@gmail.com<br>8574 8-Bit I/O Expander with                           | a Serial Interface with your Arduino This                                      | s library provides an easier interface to |
| control the PCF8   | 3574.   |  | v   |

#### 18.1.1. IoTPSP\_PCF8574\_GET\_ADDRESS

Come detto in precedenza, a seconda della disponibilità dei componenti, il PCF8574 potrebbe essere montato nel modello PCF8574 o nella sua verisone PCF8574A.

Entrambe le versioni funzionano allo stesso modo, l'unica differenza è l'indirizzo di base della comunicazione l<sup>2</sup>C.

Poiché AO, A1 e A2 del PCF8574 sono collegati a GND di default (tramite JP17, JP16 e JP15), l'indirizzo predefinito è impostato su 0x20 (PCF8574) o 0x38 (PCF8574A).

Se si dispone di un LCD 1602 è possibile verificare l'indirizzo del PCF8574 utilizzando gli sketch riportati nel § § 17 (l'indirizzo del PCF8574 viene visualizzato sul display).

Se non si dispone di un LCD 1602, è possibile utilizzare lo sketch IoTPSP\_PCF8574\_GET\_ADDRESS. Lo sketch funziona con schede ESP32, ESP8266 e NANO.

Lo skecth cerca indirizzi compresi tra 0x20 e 0x27 (PCF8574) o tra 0x38 e 0x3F (PCF8574A) e li visualizza sul serial monitor.

Sperimentate poi con gli esempi forniti con la libreria *PCF8574 library*.

| New            | Ctrle N                               |                                   |                                     |   | <br>1750 |
|----------------|---------------------------------------|-----------------------------------|-------------------------------------|---|----------|
| Onen           | Chill D                               |                                   |                                     |   | 122      |
| Open Recent    |                                       |                                   |                                     |   |          |
| Shetchhonk     |                                       |                                   |                                     |   |          |
| Examples       |                                       | *                                 |                                     |   |          |
| Close          | Ctrl+W                                | 11.ArduinoISP                     | >                                   |   |          |
| Save           | Ctrl+S                                | Frank and the state of the second | i-o-excander-fast-easy-usage/       |   |          |
| Save As        | Ctrl+Shift+S                          | Examples for any board            | x                                   |   |          |
| Jure Han       | Curromitro                            | Adatruit Circuit Playground       | ×                                   |   |          |
| Page Setup     | Ctrl+Shift+P                          | Bridge                            | <u></u>                             |   |          |
| Print          | Ctrl+P                                | tthernet                          |                                     |   |          |
| Preferences    | Ctrl+ Comma                           | Firmata                           | 2                                   |   |          |
|                | cur commu                             | LiquidUrystal                     | >                                   |   |          |
| Quit           | Ctrl+Q                                | SD                                | >                                   |   |          |
| in scoop ()    |                                       | Stepper                           | >                                   |   |          |
| Serial.begin   | n (115200);                           | Temboo                            | >                                   |   |          |
| delay(1000);   | 7                                     | RETIRED                           | >                                   |   |          |
| // Set ninM    | de to OTTRIT                          | Examples for Arduino Nano Every   |                                     |   |          |
| pcf8574.pinl   | Hode (F0, OUTP                        | FEPROM                            | >                                   |   |          |
| pcf8574.pink   | Mode (Pl, INEU                        | SoftwareSerial                    | >                                   |   |          |
|                |                                       | SPI                               |                                     |   |          |
| if (pcf8574    | beats())/                             | Wire                              | Arduino4Leds4ButtonsWithInterrupt   |   |          |
| Serial pr      | intin("OK");                          |                                   | blinkOnPin0                         |   |          |
| ]else{         | Contraction of the second second      | Examples from Custom Libraries    | encoderWithBasicLibraryFunction     |   |          |
| Serial.pr:     | intin("RO");                          | ACROBOTIC SSD1306                 | encoderWithFullLibraryFunction      |   |          |
| 1              |                                       | Adafruit BusiO                    | interruptWemos                      |   |          |
|                |                                       | Adafruit GFX Library              | keyPressed_withInterrupt            |   |          |
| id loop()      |                                       | Adafruit SSD1306                  | keyPressedPin1                      |   |          |
| nof8574 dia    | tal Marine ( DD                       | Adafruit Unified Sensor           | keyPressedPin1_storedOnBuffer_async |   |          |
| delay (1000) : | i i i i i i i i i i i i i i i i i i i | DFRobotDFPlayerMini               | ledEsp32OnTheSecondI2C              |   |          |
| pof8574.dig    | italWrite (PO,                        | DHT sensor library                | ledWernos                           |   |          |
| delay(1000);   | ;                                     | Grove 4-Digit Display             | readAll_Interrupt                   |   |          |
|                |                                       | LiquidCrystal I2C                 | readAll_Interrupt_IowMemory         |   |          |
|                |                                       | PCF8574 library                   | writeAllEsp8266                     |   |          |
|                |                                       | RICIB                             | 2                                   |   |          |
|                |                                       | Servo                             | >                                   | _ | <br>_    |
|                |                                       | StepperDriver                     | >                                   |   |          |
|                |                                       | TinyWireM                         | >                                   |   |          |
|                |                                       | TM1637                            | 5                                   |   |          |
|                |                                       |                                   |                                     |   |          |

# 19. TM1637 Digital display

La **IOT PROTO SHIELD PLUS** è predisposta per lavorare con display a sette segmenti TM1367 a 4pin, largamente diffusi sul mercato.

Per lavorare con un display a sette segmenti TM1367, è sufficiente collegare il modulo all'header J20 come mostrato nella Figure 40.



Figure 38 – Header J20 dedicato al display a sette segmenti sulla IDT PROTO SHIELD PLUS



Figure 39 – Schema connessioni dell'header J20 della IoT PROTO SHIELD PLUS

IoT Proto Shield Plus Manuale Utente



Figure 40 – Modulo TM1367 inserito nell'header J20 sulla IOT PROTO SHIELD PLUS

### 19.1. Connessioni di default

Come mostrato in Figure 39, la sezione TM1637 della IoT Ркото Sнield Plus usa i segnali del bus I<sup>2</sup>C per pilotare il chip TM1637.

In alcuni casi, questo può creare dei conflitti con altre periferiche I<sup>2</sup>C se utilizzate contemporaneamente al TM1637.

Si veda §14.1 relativo ai segnali del bus l<sup>2</sup>C sulla IOT PROTO SHIELD PLUS.

#### 19.2. Sample Sketches

Posizione: IoTPSP Sample Sketches\IoTPSP TM1637 Display

Gli scketch di esempio sono basati sul tutorial al link: <u>https://create.arduino.cc/projecthub/ryanchan/tm1637-digit-display-arduino-quick-tutorial-ca8a93</u>

Gli sketch sono stati adattati ai segnali del bus I<sup>2</sup>C della IOT PROTO SHIELD PLUS (vedi §14.1).

Gli sketch fanno uso delle librerie *TM1637* e *Grove 4-Digit Display* libraries. Assicuratevi di averle installate prima di utilizzare gli sketch.



# www.GtronicsShop.com www.Gtronics.NET

| Dibrary Manager  | ×                 |
|--|-------------------|
|  |                   |
| implementend. On top of these regular functionality a segerate fun class which adds more features can be used. For exam<br>a bombtimer(), nightrider() and bouncingBall() method can be used when using the fun class.<br><u>More info</u>   | iple              |
| TM1637   |                   |
| by Avishay Orpaz Version 1.2.0 INSTALLED<br>Driver for 4 digit 7-segment display modules, based on the TM1637 chip. These chips can be found in cheap display modules.<br>They communicate with the processor in I2C-like protocol. The implementation is pure software emulation and doesn't ma<br>use of any special hardware (other than GPIO pins). It is assumed that pull-up resistors are present (usually integrated in<br>display module).<br>More info | les.<br>ke<br>the |
|  |                   |
| by AKJ<br>Feature-full and simple TM1637 library with nonblocking animation support Features: -Display integers, float, string with o<br>one functionNonblocking animation: Blink, scrollLeft, fadeIn, fadeOut -Customizable offset, pad, overflowCustom raw<br>displayTunable brightnessScreen clearing and splitting. Visit https://github.com/AKJ7/TM1637 for more information.<br><u>More info</u>   | nly<br>value<br>V |
|  | Close             |
|  |                   |
| Jibrary Manager  | ×                 |
| ype All v Topic All v grove 4-digit  |                   |
| More info  | ^                 |
| Grove 4-Digit Display  |                   |
| by Seeed Studio  |                   |

 Grove 4-Digit Display

 by Seeed Studio

 Arduino library to control Grove\_4Digital\_Display TM1637. 4 digit display module is usually a 12 pin module. In this Grove gadget, we utilize a TM1637 to scale down the controlling pins into 2 Grove pins. It only takes 2 digital pins of Arduino or Seedurino to control the content, even the luminance of this display. For projects that require of alpha-numeric display, this can be a nice choice.

 More info
 Image: Control Grove 6-Axis Digital Accelerometer&Cyroscope(ADIS16470). The ADIS16470 is a miniature MEMS inertial measurement unit (IMU) that includes a triaxial gyroscope and a triaxial accelerometer.

 More info
 Image: Control Grove 6-Axis Digital Accelerometer&Cyroscope(ADIS16470). The ADIS16470 is a miniature MEMS inertial measurement unit (IMU) that includes a triaxial gyroscope and a triaxial accelerometer.

Utilizzare lo sketch <code>IoTPSP\_TM1637\_Display\_NANO</code> per testare il display TM1637 con schede ARDUINO NANO.

Utilizzare lo sketch IoTPSP\_TM1637\_Display\_ESP32 per testare il display TM1637 con schede ESP32.

Utilizzare lo sketch IoTPSP\_TM1637\_Display\_ESP8266 per testare il display TM1637 con schede ESP8266.

# 19.3. Usare il display TM1637 con pin GPIO differenti

Si veda § 14.3 relativo a come utilizzare il bus l<sup>2</sup>C con pin GPIO differenti sulla IoT Proto Shield Plus

# 20. RELAY

La IoT PROTO SHIELD PLUS è predisposta per lavorare con i moduli RELÈ a 3 pin largamente diffusi sul mercato.

Per lavorare con un modulo RELÈ, inserirlo semplicemente nell'header J27.



Figure 41 – Sezione RELÈ della IoT PROTO SHIELD PLUS



Figure 42 – Schema collegamenti della sezione RELÈ della IDT PROTO SHIELD PLUS



Figure 43 – Modulo RELÈ inserito nell'header J27 della IOT PROTO SHIELD PLUS

|                | LOI OLOO |
|----------------|----------|
| RELAY GPIO12 D | NC NC    |

NC = Non Connesso

Si noti che non ci sono connessioni predefinite per schede ESP8266. Per utilizzare moduli RELÈ con schede ESP8266, è necessario eseguire i collegamenti con dei cavetti volanti.

### 20.2. Connessioni suggerite per lavorare con schede ESP8266

Gli sketch di esempio funzionano con le seguenti connessioni: J3 pin GPIOO (D3) dell'ESP8622 connesso al pin GPIO12 dell'ESP32 (RELAY). J3 pin GPIO14 (D5) dell'ESP8622 connesso al pin GPIO32 dell'ESP32 (TOUCH).



Figure 44 – Connessioni suggerite per lavorare con schede ESP8266 e moduli RELÈ

#### 20.3. Sketch di esempio

Posizione: IoTPSP\_Sample\_Sketches\IoTPSP\_RELAY

Usare lo sketch IoTPSP\_RELAY\_NANO per testare il RELÈ con schede ARDUINO NANO. Usare lo sketch IoTPSP\_RELAY\_ESP32 per testare il RELÈ con schede ESP32. Usare lo sketch IoTPSP\_RELAY\_ESP8266 per testare il RELÈ con schede ESP8266.

Questi sketch usano il sensore TOUCH per commutare il RELÈ. Toccate i cerchi del TOUCH, il RELÈ commuterà, ed il LED GRN si accenderà.

Si veda § 13 per maggiori dettagli sulla sezione TOUCH della IOT PROTO SHIELD PLUS.

## 20.4. Utilizzare moduli RELÈ con pin GPIO differenti

Come mostrato in Figure 42, il RELÈ è collegato al pin GPIO di default attraverso JP51. Per disconnettere il RELÈ dal pin GPIO di default, APRIRE JP51 (incidere nel centro). Tramite J25 e dei cavetti volanti, collegare il RELÈ ad un altro pin GPIO secondo le proprie esigenze. Verificare le connessioni default della IDT PROTO SHIELD PLUS (vedere §8) per evitare conflitti con altre connessioni predefinite.

# 21. DHTxx (sensore di umidità e temperatura)

La **IDT PROTO SHIELD PLUS** è predisposta per lavorare con moduli DHT22 o DHT11 a tre pin o con sensori DHT22 o DHT11 di largo consumo (vedi Figure 45).

Per lavorare con un modulo DHT con il sensore DHT, inserirlo semplicemente nell'header J31 (vidi Figure 46, Figure 48 e Figure 49).

I sensori DHT, se non montati su moduli, hanno quattro pin. Pin #3 è segnalato come "Non Connesso", tuttavia se collegato a GND il sensore funzionerà ugualmente.

Quindi se lavorate con sensori DHT potete collegarli i due modi: con pin #3 cllegato a GND, oppure piegare leggermente pin #3 (così non si inserirà nell'header e rimarrà disconnesso).



Figure 45 – Modulo DHT22 a tre pin (immagine di sinistra), sensore DHT22 a quattro pin (immagine di destra)

| 211<br>211<br>211<br>211<br>211<br>211<br>211<br>211<br>211<br>211 |   | 7 0 | Vin<br>D4 |   |     |    | USB<br>2596<br>BAT<br>]<br>J.54<br>132 | DC<br>TER | -DC<br>DR<br>Y SH |   |   |   | V CAL | out -<br>D | 000 | Ve<br>Ve<br>MA<br>MA<br>MA<br>MA<br>MA<br>MA<br>MA<br>MA<br>MA<br>MA<br>MA<br>MA<br>MA | 1 | 10 |   |   |   | a B |   | Vin+ | HT2 | 1<br>1608<br>8 6<br>0<br>0<br>0<br>0<br>0 | 97 M | )<br>9 ul |   | 31 JDUL | Vouit<br>E<br>Vouit<br>J20<br>J3 |     |   | 12 12 12 12 12 12 12 12 12 12 12 12 12 1 |       |     |   |   | 0 19 12 12 14 14 2 2 |   |   |   |   | 0 2702 |   |   | 0 35.63 2 4 3 |   |   | Period Anno |   | 1/P36<br>1/P35<br>1/P35<br>1/P29<br>1/P29<br>1/P26 |              | A A A A A A A A A A A A A A A A A A A | The second |
|--|---|-----|-----------|---|-----|----|--|-----------|-------------------|---|---|---|-------|------------|-----|--|---|----|---|---|---|-----|---|------|-----|---|------|-----------|---|---------|----------------------------------|-----|---|--|-------|-----|---|---|----------------------|---|---|---|---|--------|---|---|---------------|---|---|-------------|---|--|--------------|---------------------------------------|---|
|  | ٦ |     |           | - |     |    |  |           |                   |   | 1 | - |       |            | ŝ   | -  |   |    | Ē | 1 |   | t   |   | Ē    | -   |   | t    | -         | i |         | ľ                                |     | ŝ | -  |       |     | - |   |                      |   | Ē | - |   |        | - | ŝ |               |   |   | i           |   |  |              | 11                                    |   |
| E.C.9  | 1 |     |           | - |     |    | -                                      |           |                   |   |   |   |       |            | i   |  |   |    | - |   |   | t   | ľ | -    | •   |   |      |           | - |         |                                  |     |   |  |       |     | : |   |                      | ï | - |   |   |        | - |   |               |   |   | -           |   |  | and a second |                                       |   |
| 129<br>GRB   | 4 | -   |           | - |     |    | -                                      |           |                   |   | - | • | -     |            | 1   | •  | 1 | -  | * | • |   |     |   | •    | 1   |   |      | -         | • |         |                                  |     | * | •  |       |     | * | • | • •                  |   | • | • |   |        | * | * |               |   |   | -           |   |  | of Bars      | ()<br>()                              |   |
| GHD<br>GHD   | Т |     |           |   |     |    |  |           |                   |   |   |   |       |            |     |  |   |    |   |   |   |     |   |      |     |   |      |           |   |         |                                  |     |   |  | <br>  |     |   |   |                      |   |   |   |   |        |   |   |               |   |   |             |   |  |              | 10                                    | 0   |
| - pro-   |   | -   |           |   |     |    |  |           |                   |   |   |   |       |            |     |  |   |    |   |   |   | 1   |   |      |     |   |      |           |   |         |                                  |     |   |  |       |     |   |   |                      |   |   |   |   |        |   |   |               |   |   |             |   |  |              | 20                                    | 5   |
| DHT  |   |     |           |   |     | ł. |  |           | ł                 | ÷ |   |   |       | ÷          | 2   |  |   | 1  | - | 2 |   | ł   | h | -    |     |   | ÷    | -         | - |         |                                  | ł   | - |  | 2     |     | : |   |                      |   | - | - |   |        | - | - |               |   |   | -           |   |  |              |                                       | 199   |
| BRDG   | J |     |           | * |     |    |  |           |                   | - |   |   |       |            |     |  |   |    |   |   |   |     |   |      |     |   |      |           |   |         |                                  |     |   |  |       |     |   |   |                      |   |   |   |   |        |   |   |               |   |   |             |   |  | -            |                                       | 1000  |
| 00   | - | _   | -         | - | _   | -  | -                                      | _         | _                 | _ |   | - | _     | -          | _   | _  | _ |    |   |   | _ | 3   | - | _    | _   | -   | _    | _         | _ | _       | -                                | _   | _ | _  | <br>_ | _   | _ | _ | _                    | _ | _ |   |   | _      | _ | _ | -             | _ | _ | -           | _ | -  |              | -                                     |   |
|  |   |     |           |   |     |    | 1                                      |           |                   |   |   | 1 |       |            |     |  |   |    |   |   |   |     |   |      | * 1 |   |      |           |   |         |                                  |     |   |  | <br>  | 0.1 |   |   |                      |   |   |   |   |        |   |   |               |   |   |             |   |  |              | AU<br>E DUI                           |   |
| CHE CHE  |   |     | *         | * | • • |    | 3                                      |           |                   |   |   | 1 |       |            | *   |  |   |    |   |   |   | *   |   | *    | * 1 | •   |      | *         | * |         | • •                              | ti. |   |  | <br>  | 0   | * |   |                      |   |   |   |   |        | * |   | * *           |   |   |             |   |  | Ĩ            | 110                                   | ő   |
| O RST  | - | _   | -         | - | -   | -  | -                                      | -         | -                 | - | - | - | -     | -          | -   | -  | - | _  | - | - | - | -   | _ | -    | -   | -   | -    | -         | - | -       | -                                | -   |   |  |       |     |   |   | -                    |   |   |   | - | -      | - |   |               |   |   |             |   |  | -            |                                       |   |

Figure 46 – Sezione DHT della IOT PROTO SHIELD PLUS

# www.GtronicsShop.com www.Gtronics.NET





Figure 47 – Schema connessioni della sezione DHT della IoT PROTO SHIELD PLUS



Figure 48 – Modulo DTH22 inserito in J31 della IOT PROTO SHIELD PLUS



Figure 49 – Sensore DTH22 inserito in J31 della IDT PROTO SHIELD PLUS

#### 21.1. Connessioni di default

|     | ESP32  | ARDUINO | ESP8266 |
|-----|--------|---------|---------|
| DHT | GPIO13 | D9      | NC      |

*NC* = Non Connesso

Si noti che non ci sono connessioni predefinite per schede ESP8266. Per utilizzare sensori e /o moduli DHT con schede ESP8266, è necessario eseguire i collegamenti con dei cavetti volanti.

## 21.2. Connessioni suggerite per lavorare con schede ESP8266

Gli sketch di esempio funzionano con le seguenti connessioni: J3 pin GPIO0 (D3) dell'ESP8622 connesso al pin GPIO13 dell'ESP32 (DHT).



Figure 50 – Connessioni suggerite per lavorare con schede ESP8266 e moduli o sensori DHT

## 21.3. Sketch di esempio

Posizione: IoTPSP Sample Sketches\IoTPSP DHT

Gli sketch di esempio funzionano sia con moduli DHT22 che con sensori DHT22 (se utilizzte sensori DHT11 è necessario dire al software che si sta usando un sensore di tipo DHT11).

#### 21.3.1. IoTPSP\_DHT\_NANO

Lo sketch IOTPSP\_DHT\_NANO usa la libreria *DHT sensor library*, assicuratevi di averla installata prima di utilizzare lo sketch.



| 🕽 Library Manager  |            |
|--|------------|
| ype All v Topic All v dht  |            |
| Arduino library for AM2320 AM2321 and AM2323 I2C temperature and humidity sensor. Supports AM2320, AM3231 These sensors are similar to DHT12 with I2C interface.<br>More info                                  | L, AM2322. |
| DHT sensor library<br>by Adafruit<br>Arduino library for DHT11, DHT22, etc Temp & Humidity Sensors Arduino library for DHT11, DHT22, etc Temp & Hun<br>Sensors<br><u>More info</u>                             | nidity     |
| Version 1.4.1 V  | Instal     |
| by beegee_tokyo<br>Arduino ESP library for DHT11, DHT22, etc Temp & Humidity Sensors Optimized libray to match ESP32 requirements<br>changes: Back to working version by removing the last commit<br>More info | s. Last    |
| DHT12  |            |
|  | Close      |

Utilizzare lo sketch IoTPSP\_DHT\_NANO per testare moduli o sensori DHT22 con schede ARDUINO NANO. Aprire il serial monitor per visualizzare i valori di umidità e temperatura letti dal sensore DHT22.

#### 21.3.2. IoTPSP\_DHT\_ESP32

Lo sketch IOTPSP\_DHT\_ESP32 usa la libreria *DHT sensor library for ESPx*, assicuratevi di averla installata prima di utilizzare lo sketch.

| Edit Sketch   Topic   Auto Format   Archive Sketch   Fits Encoding & Reload   Manage Libraries   Ctrl+Shift+I   Serial Monitor   Ctrl+Shift+I   Pe All  Topic All  Topic All  OHT sensor library by Adafruit  Arduino library for DHT11, DHT22, etc Temp & Humidity Sensors Arduino library for DHT11, DHT22, etc Temp & Humidity  Sensors  Marce Info  OHT sensor library for DHT11, DHT22, etc Temp & Humidity Sensors Optimized libray to match ESP32 requirements. Last  changes: Back to working version by removing the last commit  More Info  Outside  Outside  Topic DHT12 temperature and humidity sensor. DHT12  More Info  More Info  | IoTPSP_SSD130  |  |   |   |                                     |                                     |  |                      |   |
|---|--|--|---|---|-------------------------------------|-------------------------------------|--|----------------------|---|
| Auto Format Ctrl+T<br>Archive Sketch<br>Fix Encoding & Reload<br>Autois Sketch<br>Fix Encoding & Reload<br>Autois is an<br>be sketch in Senal Monitor Ctrl+Shift+H<br>Library Manager<br>pe Al Topic Al Top  | e Edit Sketch T  | fools Help   |   |   |                                     |                                     |  |                      |   |
| Archive Sketch<br>Fix Encoding & Reload<br>Manage Libraries<br>Library Manager<br>pe Al v Topic Al v dht<br>DHT sensor library<br>by Adafruit<br>Arduino library for DHT11, DHT22, etc Temp & Humidity Sensors Arduino library for DHT11, DHT22, etc Temp & Humidity<br>Sensors<br>Mere Life<br>DHT sensor library for ESPX<br>by beege_tokyo<br>Arduino ESP library for DHT11, DHT22, etc Temp & Humidity Sensors Optimized libray to match ESP32 requirements. Last<br>changes: Back to working version by removing the last commit<br>More info<br>Present<br>Arduino library for I2C DHT12 temperature and humidity sensor. DHT12<br>More info  |  | Auto Format  | C   | Ctrl+T  |                                     |                                     |  |                      |   |
| oTPSP_SSD1 Fix Encoding & Reload  | ~  | Archive Sketch   |   |   |                                     |                                     |  |                      |   |
| Manage Libraries       Ctrl+Shift+I         his is an each of the sketch m       Serial Monitor         Ctrl+Shift+M         Library Manager         pe [Al]       Topic [Al]         DHT sensor library         by Adafruit         Arduino library for DHT11, DHT22, etc Temp & Humidity Sensors Arduino library for DHT11, DHT22, etc Temp & Humidity Sensors         DHT sensor library for ESPx         by beegee_tokyo         Arduino SP library for DHT11, DHT22, etc Temp & Humidity Sensors Optimized libray to match ESP32 requirements. Last changes: Back to working version by removing the last commit         More_info         Version 1.17.0       Install         OHT12         by Rob Tillaart         Arduino library for 12C DHT12 temperature and humidity sensor. DHT12         More_info   | oTPSP_SSD1   | Fix Encoding & Reload  |   |   |                                     |                                     |  |                      |   |
| DHI seasor library   by Adafruit   Arduino library for DHT11, DHT22, etc Temp & Humidity Sensors Arduino library for DHT11, DHT22, etc Temp & Humidity Sensors Arduino library for DHT11, DHT22, etc Temp & Humidity Sensors Arduino library for DHT11, DHT22, etc Temp & Humidity Sensors Arduino library for DHT11, DHT22, etc Temp & Humidity Sensors Arduino library for DHT11, DHT22, etc Temp & Humidity Sensors Optimized library to match ESP32 requirements. Last changes: Back to working version by removing the last commit   More info   Version 1.17.0   Utstall  | ******   | Manage Libraries   | C   | Ctrl+Shift+I  |                                     |                                     |  |                      |   |
| Library Manager<br>pe All ✓ Topic All ✓ dht<br>DHT sensor library<br>by Adafruit<br>Arduino library for DHT11, DHT22, etc Temp & Humidity Sensors Arduino library for DHT11, DHT22, etc Temp & Humidity<br>Sensors<br>there lefe<br>DHT sensor library for ESPx<br>by beegee_tokyo<br>Arduino ED library for DHT11, DHT22, etc Temp & Humidity Sensors Optimized libray to match ESP32 requirements. Last<br>changes: Back to working version by removing the last commit<br>More info<br>DHT2<br>by Rob Tillaart<br>Arduino library for I2C DHT12 temperature and humidity sensor. DHT12<br>More info  | his is an he sketch m  | Serial Monitor   | C   | trl+Shift+M   |                                     |                                     |  |                      |   |
| pe All v Topic All dht<br>DHT sensor library<br>by Adafruit<br>Arduino library for DHT11, DHT22, etc Temp & Humidity Sensors Arduino library for DHT11, DHT22, etc Temp & Humidity<br>Sensors<br>More table<br>DHT sensor library for ESPx<br>by begee_tokyo<br>Arduino ESP library for DHT11, DHT22, etc Temp & Humidity Sensors Optimized libray to match ESP32 requirements. Last<br>changes: Back to working version by removing the last commit<br>More info<br>Version 1.17.0 Install<br>FILE<br>by Rob Tillaart<br>Arduino library for I2C DHT12 temperature and humidity sensor. DHT12<br>More info   | Library Mana   | ger  |   |   |                                     |                                     |  |                      |   |
| DHT sensor library<br>by Adafruit<br>Arduino library for DHT11, DHT22, etc Temp & Humidity Sensors Arduino library for DHT11, DHT22, etc Temp & Humidity<br>Sensors<br>What sensor library for ESPx<br>by begge_tokyo<br>Arduino ESP library for DHT11, DHT22, etc Temp & Humidity Sensors Optimized libray to match ESP32 requirements. Last<br>changes: Back to working version by removing the last commit<br>More info<br>Version 1.17.0 Version 1.17.0 Install<br>Arduino library for I2C DHT12 temperature and humidity sensor. DHT12<br>More info  | pe All   | V Topic All  | √ dht   |   |                                     |                                     |  |                      |   |
| HT sensor library<br>by Adafruit<br>Arduino library for DHT11, DHT22, etc Temp & Humidity Sensors Arduino library for DHT11, DHT22, etc Temp & Humidity<br>Sensors<br>HT sensor library for ESPx<br>by begge_tokyo<br>Arduino ESP library for DHT11, DHT22, etc Temp & Humidity Sensors Optimized libray to match ESP32 requirements. Last<br>thanges: Back to working version by removing the last commit<br>More info<br>Version 1.17.0 Version 1.17.0 Install<br>H12<br>by Rob Tillaart<br>Arduino library for I2C DHT12 temperature and humidity sensor. DHT12<br>Agree info  |  |  |   |   |                                     |                                     |  |                      |   |
| HT sensor library<br>by Adafruit<br>Arduino library for DHT11, DHT22, etc Temp & Humidity Sensors Arduino library for DHT11, DHT22, etc Temp & Humidity<br>Sensors<br>Were tafe<br>HT sensor library for ESPx<br>by beegee_tokyo<br>Arduino ESP library for DHT11, DHT22, etc Temp & Humidity Sensors Optimized libray to match ESP32 requirements. Last<br>thanges: Back to working version by removing the last commit<br>Wore info<br>Version 1.17.0 ↓ Install<br>HT12<br>by Rob Tillaart<br>Arduino library for I2C DHT12 temperature and humidity sensor. DHT12<br>Arduino library for I2C DHT12 temperature and humidity sensor. DHT12<br>Arduino library for I2C DHT12 temperature and humidity sensor. DHT12<br>Arduino library for I2C DHT12 temperature and humidity sensor. DHT12  |  |  |   |   |                                     |                                     |  |                      |   |
| Arduino library for DHT11, DHT22, etc Temp & Humidity Sensors Arduino library for DHT11, DHT22, etc Temp & Humidity<br>Sensors<br>HTT sensor library for ESPx<br>by begee_tokyo<br>Arduino ESP library for DHT11, DHT22, etc Temp & Humidity Sensors Optimized libray to match ESP32 requirements. Last<br>changes: Back to working version by removing the last commit<br>More info<br>Version 1.17.0 V<br>Instal<br>Wresion 1.17.0 V<br>Instal<br>Arduino library for I2C DHT12 temperature and humidity sensor. DHT12<br>More info   | )HT sensor lib   | rary   |   |   |                                     |                                     |  |                      |   |
| Sensors<br>Munit July<br>DHT sensor library for ESPx<br>by begee_tokyo<br>Arduino ESP library for DHT11, DHT22, etc Temp & Humidity Sensors Optimized libray to match ESP32 requirements. Last<br>changes: Back to working version by removing the last commit<br>More info<br>Version 1.17.0 V<br>Instal<br>OHT12<br>by Rob Tillaart<br>Arduino library for I2C DHT12 temperature and humidity sensor. DHT12<br>More info  | DHT sensor lib<br>by Adafruit  | rary   |   |   |                                     |                                     |  |                      |   |
| DHT sensor library for ESPx<br>by beegee_tokyo<br>Arduino ESP library for DHT11, DHT22, etc Temp & Humidity Sensors Optimized libray to match ESP32 requirements. Last<br>changes: Back to working version by removing the last commit<br>More info<br>Version 1.17.0 V<br>Install<br>of the sensor of the sens | DHT sensor lib<br>by Adafruit<br>Arduino librar  | rary<br>ry for DHT11, DHT22, etc T   | femp & Humidity Se  | <b>ensors</b> Arduing                                     | library for DHT                     | 11, DHT22, (                        | etc Temp & H                                   | umidity              |   |
| HT sensor library for ESPx         by beegee_tokyo         Arduino ESP library for DHT11, DHT22, etc Temp & Humidity Sensors Optimized libray to match ESP32 requirements. Last changes: Back to working version by removing the last commit         More info         Version 1.17.0 v         Install         Version 1.17.0 v         MILL         by Rob Tillaart         Arduino library for I2C DHT12 temperature and humidity sensor. DHT12  | OHT sensor lib<br>by Adafruit<br>Arduino librar<br>Sensors   | rary<br>ry for DHT11, DHT22, etc T   | femp & Humidity Se  | <b>ensors</b> Arduind                                     | library for DHT                     | 11, DHT22, e                        | etc Temp & H                                   | umidity              |   |
| by beegee_tokyo<br>Arduino ESP library for DHT11, DHT22, etc Temp & Humidity Sensors Optimized libray to match ESP32 requirements. Last<br>changes: Back to working version by removing the last commit<br>More info<br>Version 1.17.0 V<br>Install<br>by Rob Tillaart<br>Arduino library for I2C DHT12 temperature and humidity sensor. DHT12<br>More info   | DHT sensor lib<br>by Adafruit<br>Arduino librar<br>Sensors   | rary<br>ry for DHT11, DHT22, etc T   | ſemp & Humidity So  | <b>ensors</b> Arduinc                                     | library for DHT                     | 11, DHT22, e                        | etc Temp & H                                   | umidity              |   |
| Arduino ESP library for DHT11, DHT22, etc Temp & Humidity Sensors Optimized libray to match ESP32 requirements. Last<br>changes: Back to working version by removing the last commit<br>Version 1.17.0 VInstall<br>Install<br>by Rob Tillaart<br>Arduino library for I2C DHT12 temperature and humidity sensor. DHT12<br>More info  | OHT sensor lib<br>by Adafruit<br>Arduino librar<br>Sensors<br>Marce 1969<br>DHT sensor lib   | rary<br>ry for DHT11, DHT22, etc T<br>rary for ESPx  | ſemp & Humidity So  | <b>ensors</b> Arduinc                                     | library for DHT                     | 11, DHT22, 4                        | etc Temp & H                                   | umidity              |   |
| Version 1.17.0 Version by removing the last commit<br>Version 1.17.0 Version 1.17.0 Install<br>by Rob Tillaart<br>Arduino library for I2C DHT12 temperature and humidity sensor. DHT12<br>More info   | OHT sensor lib<br>by Adafruit<br>Arduino librar<br>Sensors<br>Marce info<br>OHT sensor lib<br>by beegee_to   | rary<br>ry for DHT11, DHT22, etc T<br>rary for ESPx<br>skyo  | Temp & Humidity So  | ensors Arduine  | library for DHT                     | 11, DHT22, 4                        | etc Temp & H                                   | umidity              |   |
| Version 1.17.0 Version 1.17.0 Install<br>by Rob Tillaart<br>Arduino library for I2C DHT12 temperature and humidity sensor. DHT12<br>More info   | DHT sensor lib<br>by Adafruit<br>Arduino librar<br>Sensors<br>DHT sensor lib<br>by beegee_to<br>Arduino ESP l  | rary<br>ry for DHT11, DHT22, etc T<br>rary for ESPx<br>skyo<br>library for DHT11, DHT22, /   | Temp & Humidity So<br>etc Temp & Humidi   | ensors Arduino<br>ity Sensors Op                          | library for DHT                     | 11, DHT22, 4                        | atc Temp & H<br>32 requireme                   | umidity<br>nts. Last |   |
| by Rob Tillaart<br>Arduino library for I2C DHT12 temperature and humidity sensor. DHT12<br>More info  | DHT sensor lib<br>by Adafruit<br>Arduino librar<br>Sensors<br>Mara 165<br>DHT sensor lib<br>by beegee_to<br>Arduino ESP I<br>changes: Bad<br>More info               | rary<br>ry for DHT11, DHT22, etc T<br>rary for ESPx<br>skyo<br>library for DHT11, DHT22,<br>k to working version by rer                                    | Temp & Humidity So<br>etc Temp & Humidi<br>moving the last con                        | <b>ensors</b> Arduino<br><b>ity Sensors</b> Op<br>nmit    | library for DHT<br>imized libray to | 11, DHT22, 4                        | atc Temp & H<br>32 requireme                   | umidity<br>nts. Last |   |
| by Rob Tillaart<br>Arduino library for I2C DHT12 temperature and humidity sensor. DHT12<br>More info  | DHT sensor lib<br>by Adafruit<br>Arduino librar<br>Sensors<br>Mana lab<br>DHT sensor lib<br>by beegee_to<br>Arduino ESP I<br>changes: Bacl<br>More info              | rary<br>ry for DHT11, DHT22, etc T<br>rary for ESPx<br>skyo<br>library for DHT11, DHT22,<br>k to working version by rer                                    | Temp & Humidity So<br>etc Temp & Humidi<br>moving the last con                        | <b>ensors</b> Arduino<br><b>ity Sensors</b> Op<br>nmit    | library for DHT<br>imized libray to | 11, DHT22, 4<br>o match ESP:        | etc Temp & H<br>32 requireme                   | umidity<br>nts. Last |   |
| by Rob Tillaart<br>Arduino library for I2C DHT12 temperature and humidity sensor. DHT12<br><u>More info</u>   | DHT sensor lib<br>by Adafruit<br>Arduino librar<br>Sensors<br>Marce 1afe<br>MHT sensor lib<br>by beegee_to<br>Arduino ESP I<br>changes: Bacl<br>More info            | rary<br>ry for DHT11, DHT22, etc T<br>rary for ESPx<br>skyo<br>library for DHT11, DHT22,<br>k to working version by rer                                    | Femp & Humidity So<br>etc Temp & Humidi<br>moving the last con                        | <b>ensors</b> Arduind<br><b>ity Sensors</b> Op<br>nmit    | library for DHT                     | 11, DHT22, 4<br>o match ESP:<br>Ve  | etc Temp & H<br>32 requireme<br>rsion 1.17.0 v | umidity<br>nts. Last |   |
| Arduno library for 12C DH122 temperature and numidity sensor. DH112 <u>More info</u>  | DHT sensor lib<br>by Adafruit<br>Arduino librar<br>Sensors<br>Marce 166<br>DHT sensor lib<br>by beegee_to<br>Arduino ESP I<br>changes: Back<br>More info             | rary<br>ry for DHT11, DHT22, etc T<br>rary for ESPx<br>skyo<br>library for DHT11, DHT22,<br>k to working version by rer                                    | Femp & Humidity So<br>etc Temp & Humidi<br>moving the last con                        | <b>ensors</b> Arduino<br><b>ity Sensors</b> Op<br>nmit    | library for DHT<br>imized libray to | 11, DHT22, o<br>o match ESP:<br>Vei | atc Temp & H<br>32 requireme<br>rsion 1.17.0 v | umidity<br>nts. Last |   |
|   | DHT sensor lib<br>by Adafruit<br>Arduino librar<br>Sensors<br>Marce 156<br>DHT sensor lib<br>by beegee_to<br>Arduino ESP I<br>changes: Back<br>More info             | rary<br>ry for DHT11, DHT22, etc T<br>rary for ESPx<br>skyo<br>library for DHT11, DHT22,<br>k to working version by rer                                    | Temp & Humidity So<br>etc Temp & Humidi<br>moving the last con                        | ensors Arduind<br>ity Sensors Op<br>nmit                  | library for DHT<br>imized libray to | 11, DHT22, d<br>o match ESP:<br>Ve  | atc Temp & H<br>32 requireme<br>rsion 1.17.0 v | umidity              |   |
|   | DHT sensor lib<br>by Adafruit<br>Arduino librai<br>Sensors<br>Sensors<br>Marce iafe<br>DHT sensor lib<br>by beegee_to<br>Arduino ESP I<br>changes: Back<br>More info | rary<br>ry for DHT11, DHT22, etc T<br>rrary for ESPx<br>skyo<br>library for DHT11, DHT22,<br>k to working version by rer<br>t<br>ry for 12C DHT12 temperal | Temp & Humidity So<br>etc Temp & Humidi<br>moving the last con<br>ture and humidity : | ensors Arduind<br>ity Sensors Op<br>nmit<br>sensor. DHT12 | library for DHT<br>imized libray to | 11, DHT22, d<br>o match ESP:<br>Ve  | atc Temp & H<br>32 requireme<br>rsion 1.17.0 v | umidity<br>nts. Last | ] |

Utilizzare lo sketch <code>IoTPSP\_DHT\_ESP32</code> per testare moduli o sensori DHT22 con schede ESP32.

Aprire il serial monitor per visualizzare i valori di umidità e temperatura letti dal sensore DHT22.

#### 21.3.3. IoTPSP\_DHT\_ESP8266

Lo sketch IOTPSP\_DHT\_ESP32 usa la libreria *DHT sensor library for ESPx*, assicuratevi di averla installata prima di utilizzare lo sketch (vedi § 21.3.2).

Utilizzare lo sketch IoTPSP\_DHT\_ESP8266 per testare moduli o sensori DHT22 con schede ESP8266. Aprire il serial monitor per visualizzare i valori di umidità e temperatura letti dal sensore DHT22.

### 21.4. Utilizzare moduli e sensori DTHxx con pin GPIO differenti

Come mostrato in Figure 47, il segnale dati dei moduli o sensori DHTxx è collegato al pin GPIO di default attraverso JP56.

Per disconnettere il segnale dati dal pin GPIO di default, APRIRE JP56 (incidere nel centro).

Tramite J29 e dei cavetti volanti, collegare il segnale dati ad un altro pin GPIO secondo le proprie esigenze. Verificare le connessioni default della IDT PROTO SHIELD PLUS (vedere §8) per evitare conflitti con altre connessioni predefinite.

# 22. RTC (Real Time Clock)

La **IDT PROTO SHIELD PLUS** è predisposta per lavorare con moduli RTC l<sup>2</sup>C largamente diffusi sul mercato (vedi Figure 51).

É possible lavorare sia con moduli DS3231 che DS1307. Si suggerisce l'utilizzo di moduli DS3231 che sono molto stabili: la deriva temporale è davvero bassa anche su lunghi periodi di tempo.

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.

Per lavorare con un modulo RTC, basta inserirlo nell'header J11.

J11 è un header a quattro pin, controllare i segnali sul modulo RTC per assicurarsi che corrispondano alla piedinatura di J11 (GND, Vcc, SDA e SCL).

Alcune schede (come Arduino NANO33 ed ESP32) hanno un proprio RTC interno, questa sezione riguarda l'utilizzo dei moduli DS3231 o DS1307.



Figure 51 – Moduli RTC di commune utilizzo, DS3231 (sinistra), DS1307 (destra)



IOT PROTO SHIELD PLUS MANUALE UTENTE



Figure 52 – Header J11 della IOT PROTO SHIELD PLUS



Figure 53 – Schema connessioni dell'header J11 della IoT PROTO SHIELD PLUS



Figure 54 – Modulo RTC DS3231 inserito nell'header J11 della IOT PROTO SHIELD PLUS

Rev.1.07.i

Solitamente, questi moduli sono dotati di un porta batteria e di un circuito di carica della batteria. Se si utilizza una batteria NON ricaricabile, si consiglia di "disabilitare" il circuito di ricarica per non danneggiare la batteria.

Il modo più semplice per "disabilitarlo" è rimuovere il diodo del circuito di carica (vedi Figure 55).



Figure 55 – Il diodo del circuito di carica della batteria di un DS3231 (sinistra), e di una DS1307 (destra)

### 22.1. Connessioni di default

Si veda §14.1 relativo ai segnali del bus l<sup>2</sup>C sulla IOT PROTO SHIELD PLUS.

### 22.2. Sketch di esempio

Posizione: IoTPSP Sample Sketches\IoTPSP RTC

Gli sketch usano la libreria *RTClib* library di Adafruit. Assicuratevi di averla installata prima di utilizzare gli sketch.



| 💿 Library Manager   | × |
|---|---|
| Type All V Topic All V rtdib  |   |
| D53231  | ^ |
| by Andrew Wickert, Eric Ayars, Jean-Claude Wippler, Northern Widget LLC<br>Arduino library for the DS3231 real-time clock (RTC) Abstracts functionality for clock reading, clock setting, and alarms for the<br>DS3231 high-precision real-time clock. This is a splice of Ayars' (http://hacks.ayars.org/2011/04/ds3231-real-time-clock.html)<br>and Jeelabs/Ladyada's (https://github.com/adafruit/RTClib) libraries.<br><u>More info</u> |   |
| RTClib<br>by Adafruit Version 1.12.4 INSTALLED<br>A fork of Jeelab's fantastic RTC library A fork of Jeelab's fantastic RTC library<br>More info<br>Select version V Instal   |   |
| RTCLib by NeiroN  |   |
| by JeeLabs (http://news.jeelabs.org/code/), NeiroN (neiron.nxn@gmail.com)<br>A library that makes interfacing DS1302, DS1307, DS3231, PCF8583, PCF8563, RTC_Millis Real Time Clock modules easy.<br>Including temperature, alarms and memory storage if present. Includes DateTime class implementation and it conversion.<br>More info   |   |
|   | ¥ |
| Clos  | æ |

Usare lo sketch IoTPSP\_RTC\_NANO per testare i moduli RTC con schede ARDUINO NANO. Usare lo sketch IoTPSP\_RTC\_ESP32 per testare i moduli RTC con schede ESP32. Usare lo sketch IoTPSP\_RTC\_ESP8266 per testare i moduli RTC con schede ESP8266.

Aprire il serial monitor per vedere la data ed ora lette dal modulo RTC.

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.

Se state usando un modulo DS3231 otterrete anche la temperatura del modulo, mentre con un modulo DS1307 i dati di temperatura non hanno significato.

## 22.3. Usare I moduli RTC con pin GPIO differenti

Si veda § 14.3 relativo a come utilizzare il bus I<sup>2</sup>C con pin GPIO differenti sulla IOT PROTO SHIELD PLUS.

# 23. Bus di comunicazione SPI sulla IOT PROTO SHIELD PLUS

SPI (Serial Peripheral Interface) è un bus di comunicazione seriale sincrono utilizzato per le comunicazioni a breve distanza, principalmente nei sistemi embedded (è possibile trovare ulteriori informazioni sul bus SPI su <u>Wikipedia</u> o altra documentazione online).



Figure 56 – Schema connessioni dei segnali del bus SPI sulla IDT PROTO SHIELD PLUS

### 23.1. Connessioni di default

|          | 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) |

Il bus SPI sulla **IDT PROTO SHIELD PLUS** è utilizzato per comunicare con moduli SD CARD (vedi § 24), o altri dispositivi per mezzo del socket SPI1 (vedi Figure 57).



Figure 57 – Socket SPI1 della IoT PROTO SHIELD PLUS

# 24. MODULO SD CARD

La **IOT PROTO SHIELD PLUS** è predisposta per lavorare con moduli SD CARD SPI a 6 pin di largo utilizzo (vedi Figure 60).

Per lavorare con un modulo SD CARD SPI, inserirlo nell'header J10 (vedi Figure 60). Naturalmente è necessario avere anche una schedina microSD inserita nel modulo.

Ai seguenti è possible trovare maggiori informazioni relative all'utilizzo delle SD card: 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 – Header J10 dedicato alla SD CARD della IOT PROTO SHIELD PLUS



Figure 59 – Schema connessione segnali dell'header J10 della IOT PROTO SHIELD PLUB



Figure 60 – Un Modulo SD CARD SPI (sinistra) ed il modulo inserito nell'header J10 della IDT PROTO SHIELD PLUS (destra)

Si veda § 23.1 relativo ai segnali del bus SPI sulla IDT PROTO SHIELD PLUS.

## 24.2. Sketch di esempio

Posizione: IoTPSP Sample Sketches\IoTPSP SD CARD

Gli sketch di esempio usano la libreria *SD library*, inclusa nell'IDE di Arduino. Gli sketch di esempio funzionano con le connessioni di default indicate al § 23.1.

#### 24.2.1. Testare la SD CARD

Questi skecth sono molto utili per testare una scheda quando non si è sicuri che essa funzioni o meno.

Utilizzare lo sketch IoTPSP\_SD\_CARD\_INFO\_NANO per visualizzare le info della SD CARD con schede ARDUINO NANO.

Utilizzare lo sketch IoTPSP\_SD\_CARD\_INFO\_per visualizzare le info della SD CARD con schede ESP32. Utilizzare lo sketch IoTPSP\_SD\_CARD\_INFO\_ESP8266 per visualizzare le info della SD CARD con schede ESP8266.

Aprire il serial monitor per visualizzare le info della SD CARD. Se il serial monitor rimane vuoto, premere il pulsante di reset.

#### 24.2.2. Leggere e scrivere su una scheda SD CARD

Una volta sicuri che la SD CARD funzioni (vedi § 24.2.1), è possibile leggere e scrivere sulla SD CARD con i seguenti sketch.

# www.GtronicsShop.com www.Gtronics.NET

Utilizzare lo sketch <code>IoTPSP\_SD\_CARD\_RW\_NANO</code> per scrivere sulla SD CARD e rileggere quanto scritto con schede ARDUINO NANO.

Utilizzare lo sketch <code>IoTPSP\_SD\_CARD\_RW\_ESP32</code> per scrivere sulla SD CARD e rileggere quanto scritto con schede ESP32.

Utilizzare lo sketch IoTPSP\_SD\_CARD\_RW\_ESP8266 per scrivere sulla SD CARD e rileggere quanto scritto con schede ESP8266.

Aprire il serial monitor per visualizzare i dati letti dalla SD CARD. Se il serial monitor rimane vuoto, premere il pulsante di reset.

# 25. LDR/xTC

La **IDT PROTO SHIELD PLUS** è predisposta per lavorare con sensori LDR, PTC o NTC. I sensori LDR (light dependent resistor, in italiano conosciuti come fotoresistenze) sono resistenze il cui valore ohmico è dipendente dalla luce (maggiori informazioni su Wikipedia).

I sensori NTC e PTC (conosciuti in italiano come termistori) sono resistenze il cui valore ohmico è dipendente dalla temperatura.

I sensori NTC hanno coefficiente negativo, mentre i sensori PTC hanno coefficiente (maggiori informazioni su <u>Wikipedia</u>).

Per lavorare con i sensori LDR, PTCo NTC è sufficiente inserirli nell'header J26.

Dato che questi sensori si comportano come resistenze variabili, il principio è quello del partitore di tensione (già visto nella sezione relativa al POTenziometro).



Figure 61 – Sezione LDR/xTC della IoT PROTO SHIELD PLUS



Figure 62 – Schema connessioni della sezione LDR/xTC of the IDT PROTO SHIELD PLUS

|         | ESP32  | ARDUINO | ESP8266 |
|---------|--------|---------|---------|
| LDR/xTC | GPIO34 | A2      | NC      |
|         |        |         |         |

*NC* = Non Connesso

Si noti che non ci sono connessioni predefinite per schede ESP8266. Per utilizzare sensori LDR/xTC con schede ESP8266, è necessario eseguire i collegamenti con dei cavetti volanti.

#### 25.2. Connessioni suggerite per lavorare con LDR/xTC con schede ESP8266

Gli sketch di esempio funzionano con le seguenti connessioni: J3 Pin A0 dell'ESP8266 connesso al pin GPIO34 dell'ESP32 GPIO34 (LDR/xTC).

Considerare che il pin A0 dell'ESP8266 è connesso di default a POT (vedere §11.1). È necessario quindi APRIRE JP12 (vedi Figure 20) per disconnettere il A0 dell'ESP8266 dal POT e lavorare correttamente quindi con i sensori LDR/xTC.



Figure 63 – Connessioni suggerite per lavorare con LDR/xTC con schede ESP8266

### 25.3. Sketch di esempio

**Posizione:** IoTPSP\_Sample\_Sketches\IoTPSP\_LDR\_xTC

A seconda della schede che sis ta utilizzando e quindi dei segnali connessi, gli sketch di esempio visualizzano sul serial monitor il valore analogico del partitore creato dal sensore LDR/xTC e la resistenza R29.

Utilizzare lo sketch IoTPSP\_LDR\_xTC\_NANO per testare sensori LDR/xTC con schede ARDUINO NANO. Utilizzare lo sketch IoTPSP\_LDR\_xTC\_ESP32 NANO per testare sensori LDR/xTC con schede ESP32. Utilizzare lo sketch IoTPSP\_LDR\_xTC\_ESP8266 NANO per testare sensori LDR/xTC con schede ESP8266.

### 25.4. Utilizzare sensori LDR/xTC con un pin GPIO differente

Come mostrato in Figure 62, il partitore creato dal sensore LDR/xTC e R29 è collegato al pin GPIO di default attraverso JP50.

Per disconnettere il partitore dal pin GPIO di default, APRIRE JP50 (incidere nel centro). Tramite J24 e dei cavetti volanti, collegare il partitore ad un altro pin GPIO secondo le proprie esigenze.

IOT PROTO SHIELD PLUS MANUALE UTENTE

Verificare le connessioni default della IOT PROTO SHIELD PLUB (vedere §8) per evitare conflitti con altre connessioni predefinite.

Il bus seriale aggiuntivo può essere utilizzato per comunicare con altri dispositivi.



Figure 64 – Schema connessioni del bus seriale aggintivo della IDT PROTO SHIELD PLUB



Figure 65 – Header J4 della IOT PROTO SHIELD PLUS

### 26.1. Connessioni di default

I segnali addizionali RX e TX sulla IoT PROTO SHIELD PLUS, sono connessi di default come segue:

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

Sulla **IDT PROTO SHIELD PLUS**, questo bus seriale addizionale è utilizzato di default per comunicare con il modulo DFPlayer Mini MP3 player (vedi §27). Eventualmente è possibile utilizzarlo per comunicare con altri dispositivi esterni tramite l'header J4 (vedi Figure 65).

# 27. DFPlayer Mini MP3 Player

La IDT PROTO SHIELD PLUS è predisposta per lavorare con moduli DFPlayer Mini MP3 ad otto pin di ampia diffusione sul mercato (vedi Figure 66).

Per lavorare con un modulo DFPlayer Mini MP3, inserirlo nel socket BRD3 (vedi Figure 66).

Vedere il seguente link per maggiori informazioni relative al modulo:

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

Per lavorare con il modulo DFPlayer Mini MP3, è necessario disporre di una scheda microSD (la scheda deve essere formattata con file system FAT32). I file MP3 devono essere in un cartella nominata "mp3".



Figure 66 – Un modulo DFPlayer Mini MP3 (sinistra) ed un modulo inserito nel socket BRD3 della IDT PROTO SHIELD PLUB (destra)



Figure 67 – Socket BRD3 della IOT PROTO SHIELD PLUS



Figure 68 – Schema connessioni del socket BRD3 della IDT PROTO SHIELD PLUS

Sulla **IDT PROTO SHIELD PLUS**, la comunicazione con il modulo DFPlayer Mini MP3 avviene mediante il bus seriale addizionale (vedi §26.1)

Il modulo DFPlayer Mini MP3 può lavorare anche a @3V3, in molti casi però i 3V3 generati dalla scheda in uso (ESP32, NANO, etc.) non è in grado di erogare la corrente necessaria al modulo per funzionare correttamente.

Quindi, come mostrato in Figure 68, il pin *Vcc* del DFPlayer Mini MP3 è connesso di default a 5V (che sono solitamente forniti dalla porta USB che state utilizzando.

### 27.2. Sketch di esempio

**Posizione:** IoTPSP\_Sample\_Sketches\IoTPSP\_MP3\_Player

Questi sketch di esempio fanno uso della libreria *DFRobotDFPlayerMini library* di DFRobot. Assicuratevi di averla installata prima di usarli.

# www.GtronicsShop.com www.Gtronics.NET

| ΙσΤ               |  |  |  |
|-------------------|--|--|--|
| PROTO SHIELD PLUS |  |  |  |
| MANUALE UTENTE    |  |  |  |

| IoTPSP_SSD1306                        | _DISPLAY_128x64_NANO   Ardu          | ino 1.8.13                           |  |    |
|---------------------------------------|--------------------------------------|--------------------------------------|--|----|
| File Edit Sketch To                   | ols 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 √                        | dfplayer                             |  |    |
|                                       |                                      |                                      |  | ^  |
|                                       |                                      |                                      |  |    |
| DFPlayerMini_Fast                     | t                                    |                                      |  | 7  |
| by PowerBroker2<br>Arduino library to | interface with the DEPlayerM         | ini MP3 module Arduino library to i  | nterface with the DEPlayerMini MP3 module      |    |
| More info                             | interface with the bir hayer         |                                      |  |    |
|                                       |                                      |                                      |  |    |
| DFRobotDFPlayer                       | Mini                                 |                                      |  | 1  |
| by DFRobot                            |                                      |                                      |  |    |
| Driver for DFPlay<br>More info        | ver Mini from DFRobot Easy-to-       | use and reliable library for DFPlay  | er Mini  |    |
|                                       |                                      |                                      | Version 1.0.5 V                                |    |
|                                       |                                      |                                      |  |    |
| T Hing Diaphay                        |                                      |                                      |  |    |
| by Gerald Lechne                      | r<br>It date, time and sensor values | as speech. The library uses the M    | 1P3 player module DEPlayer Mini for output.    |    |
| The communication                     | on with module is handled in th      | e libraray. No extra driver for DFPI | ayer is required. Only the pointer on a Serial |    |
| Interface hardwar                     | e or software is required. MP3 I     | Files for English and German are si  | upplied and need to be copied on a Micro-SD    | ~  |
|                                       |                                      |                                      | Clos   | se |

La cartella IoTPSP\_Sample\_Sketches\IoTPSP\_MP3\_Player contiene anche la cartella "mp3" contenete tre file MP3 di esempio. Copiare la cartella "mp3" sulla scheda microSD per testare gli sketch di esempio IoT PROTO SHIELO PLUS.

Questi sketch usano di default la comunicazione seriale addizionale descritta al §26.1

#### 27.2.1. IoTPSP\_MP3\_NANO

Lo sketch IoTPSP\_MP3\_NANO usa la porta seriale standard di Arduino (pin D0, D1) per comunicare con il modulo DFPlayer Mini MP3.

Dato che la seriale sarà impegnata a comunicare con il modulo MP3 Player, non potrà essere utilizzata con il Serial Monitor.

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

Lo sketch IOTPSP\_MP3\_NANO\_EVERY\_33 usa la porta di comunicazione secondaria, presente sulla schede NANO EVERY o NANO33 (i pin utilizzati sono sempre D0, D1), per comunicare con il modulo DFPlayer Mini MP3.

#### 27.2.3. IOTPSP MP3 ESP32

Lo sketch IOTPSP\_MP3\_ESP32 usa la porta di comunicazione secondaria presente sulle schede ESP32 (pin GPIO16, GPIO17), per comunicare con il modulo DFPlayer Mini MP3.
### 27.2.4. IoTPSP\_MP3\_ESP8266

Lo sketch IoTPSP\_MP3\_ESP8266\_SoftwareSerial, per testare il modulo MP3 Player con schede ESP8266, usa la libreria *EspSoftwareSerial library*, quindi assicurarsi di averla installata prima di utilizzare lo sketch.

| IoTPSP_SSD1306_DISPLAY_128x64_NANO   Arduino 1.8.13 |  |                   |       |  |  |  |  |  |  |
|---|--|-------------------|-------|--|--|--|--|--|--|
| File Edit Sketch To                                 | ols 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 Manage                                    | r  |                   | ×     |  |  |  |  |  |  |
| Type All  | ✓ Topic All ✓  | EspSoftwareSerial |       |  |  |  |  |  |  |
| EspSoftwareSeri                                     | al   |                   | ~     |  |  |  |  |  |  |
| Implementation<br>More info                         | by Dirk Kaar, Peter Lerup<br>Implementation of the Arduino software serial for ESP8266/ESP32.<br>More info |                   |       |  |  |  |  |  |  |
|   |  |                   | ~     |  |  |  |  |  |  |
|   |  |                   | Close |  |  |  |  |  |  |

The IoTPSP\_MP3\_ESP8266\_SoftwareSerial crea una porta seriale addizionale sui pin GPIOO e GPIO2, in modo da poter comunicare con il modulo DFPlayer Mini MP3.

Le schede ESP8266 hanno una porta seriale secondaria sui pin GPIO1 e GPIO3, nel caso si voglia utilizzare questa porta seriale secondaria, è necessario ridirezionare i segnali ed sperimentare autonomamente.

### 27.3. Utilizzare moduli DFPlayer Mini MP3 con pin GPIO differenti

Non è consigliabile utilizzare pin diversi per stabilire una comunicazione seriale con il modulo DFPlayer Mini MP3.

Ad ogni modo, se si sta usando una "vecchia" scheda Arduino NANO, potreste voler usare il Monitor Seriale, quindi è necessario liberare D0 e D1.

In questo caso si consiglia di usare la libreria *SoftwareSerial library* per creare una porta seriale "virtuale". Come mostrato in Figure 68, U\_RX e U\_TX sono connesso di default ai pin TX e RX del modulo DFPlayer Mini MP3, tramite JP46 e JP47.

Si può sperimentare APRENDO JP46 e JP47 (incidere al centro), reindirizzando i segnali RX e TX tramite J21 e quindi utilizzando lo sketch descritto nel § 27.3.1.

#### 27.3.1. IoTPSP\_MP3\_NANO\_SoftwareSerial

Lo sketch <code>IoTPSP\_MP3\_NANO\_SoftwareSerial</code> usa la libreria *SoftwareSerial library*, inclusa nell'IDE di Arduino.

Per utilizzare lo sketch fare riferimento al §26.1.

IOT PROTO SHIELD PLUS MANUALE UTENTE

Lo sketch utilizza D10 come pin RX e D11 come pin TX. Ulteriori informazioni sull'utilizzo della Software Serial Port sono disponibili a questo link: <u>https://www.arduino.cc/en/Reference/softwareSerial</u>.

# 28. SPEAKER

Lo speaker può essere utilizzato anche come buzzer e per creare toni modulati in frequenza. Per utilizzarlo come tone buzzer, è necessario realizzare quanto segue:

- Rimuovere il modulo DFPlayer Mini MP3 dalla socket BRD3
- Utilizzare un cavetto volante per collegare il pin #1 (Vcc) al pin #6 (SPK1) di BRD3
- Utilizzare un cavetto volante per collegare il pin #2 J21 a J23



Figure 69 – Schema connessioni necessarie all'utilizzo dello speaker come tone buzzer sulla IoT PROTO SHIELD PLUS



Figure 70 – Connessioni necessarie all'utilizzo dello speaker come tone buzzer sulla IDT PROTO SHIELD PLUS

### 28.1. Connessioni di default

Come mostrato in Figure 69, U\_TX pilota il tono del cicalino. Come mostrato nel § 26.1, U\_TX è collegato di default ai seguenti pin:

|      | ESP32  | ARDUINO | ESP8266    |
|------|--------|---------|------------|
| U_TX | GPIO17 | D1      | GPIO0 (D3) |

### 28.2. Sketch di esempio

Posizione: IoTPSP Sample Sketches \ IoTPSP SPEAKER

#### 28.2.1. IoTPSP\_SPEAKER\_NANO

Lo sketch IoTPSP\_SPEAKER\_NANO si basa sulla funzione *tone()* della libreria base di Arduino. Informazioni aggiuntive relative alla funzione *tone()* sono disponibili a questo link: <u>https://www.arduino.cc/reference/en/language/functions/advanced-io/tone/</u> La funzione non funziona(!) con schede NANO33 BLE (almeno al momento della stesura di questo manuale).

#### 28.2.2. IOTPSP\_SPEAKER\_ESP32

Lo sketch IoTPSP\_SPEAKER\_ESP32 fa uso della libreria *Tone32*. La libreria è scaricabile al link <u>https://github.com/lbernstone/Tone32</u>. Una copia della libreria è contenuta nella cartella dell'esempio.

#### 28.2.3. IOTPSP\_SPEAKER\_ESP8266

Lo sketch IoTPSP\_SPEAKER\_ESP8266 si basa sulla funzione *tone()* della libreria base di Arduino. Informazioni aggiuntive relative alla funzione *tone()* sono disponibili a questo link: <u>https://www.arduino.cc/reference/en/language/functions/advanced-io/tone/</u>

# 29. MIC

La **IDT PROTO SHIELD PLUS** è predisposta per lavorare con i moduli microfono MAX9814 o MAX4466 di ampia diffusione sul mercato.



Figure 71 – Schema connessioni della sezione MIC della IDT PROTO SHIELD PLUS



Figure 72 - Sezione MIC della IOT PROTO SHIELD PLUS

### 29.1. Modulo microfonico MAX9814

Questo modulo microfonico è basato sull'integrato MAX9814 ed è un modulo a 5 pin. Maggiori dettagli sull'integrato MAX9814 al seguente link: <u>https://datasheets.maximintegrated.com/en/ds/MAX9814.pdf</u>

This module has 5 pins:

# www.GtronicsShop.com www.Gtronics.NET

Vdd: Alimentazione modulo – 2.7-5.5 V GND: Ground OUT: Uscita analogica segnale audio Gain: Regolazione guadagno (livello di uscita) del modulo\* AR: Regolazione risoluzione\*

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

L'uscita analogica del modulo NON è "0" in assenza di suoni e rumori, l'offset è di circa 1.25V.



Figure 73 – Modulo microfonico MAX9814 di larga diffusione

Per lavorare con un modulo MAX9814, inserirlo nell'header J19, come mostrato nella figura seguente.



Figure 74 – Modulo microfonico MAX9814 inserito nell'header J19

#### 29.1.1. Impostare il guadagno (gain) del modulo

Di default il gain è impostato a 40dB: il pin #3 dell'header J19 è connesso a GND attraverso JP48. APRENDO JP48 (incidere nel mezzo) il pin #3 dell'header J19 diventa non connesso (floating), questo imposta il gain a 60dB.

Con JP48 APERTO e collegando pin #3 a Vcc, il gain viene impostato a 50dB.

#### 29.1.2. Impostare la risoluzione del modulo

Di default, la risoluzione del modulo (AR) è impostata a 1:4000: il pin #1 dell'header J19 è non connesso (floating).

By default, AR (adjusting module accuracy) is set to 1:4000: J19 pin #1 is floating.

Connettendo pin #1 dell'header J19 a Vcc, AR viene impostato a 1:2000.

Connettendo pin #1 dell'header J19 a GND, AR viene impostato a 1:500.

### 29.2. Modulo microfonico MAX4466

Questo modulo microfonico è basato sull'integrato MAX4466 ed è un modulo a 3 pin. Maggiori dettagli sull'integrato MAX4466 al seguente link: <u>https://www.maximintegrated.com/en/products/analog/audio/MAX4466.html</u>



Figure 75 – Modulo microfonico MAX4466 di larga diffusione

Per lavorare con un modulo MAX4466, inserirlo nell'header J19, come mostrato nella figura seguente. Assicuratevi che JP48 sia CHIUSO: pin #3 dell'header JP19 deve essere connesso a GND.



Figure 76 – A MAX4466 microphone plugged into J19

#### 29.3. Connessioni di default

|     | ESP32 | ARDUINO | ESP8266 |
|-----|-------|---------|---------|
| MIC | GPI39 | A7      | NC      |

*NC* = Non Connesso

Si noti che non ci sono connessioni predefinite per schede ESP8266. Per utilizzare moduli MIC con schede ESP8266, è necessario eseguire i collegamenti con dei cavetti volanti.

#### 29.4. Connessioni suggerite per lavorare con moduli MIC e schede ESP8266

Gli sketch di esempio sono fatti per lavorare con le seguenti connessioni:

Pin A0 dell'ESP8622 connesso al pin GPIO39 dell'ESP32 (MIC).

Da considerare che il pin A0 dell'ESP8266 è connesso di default al POT (vedere §11.1).

Per poter lavorare correttamente con i moduli MIC, è necessario APRIRE JP12 (vedere Figure 20) per disconnettere il pin A0 dell'ESP8266 dal POT.

|      |    |     |    |            |    |    |    |           |     |     |    |    |     |    |    |    |    | ſ  | ٦  |    |    |    |
|------|----|-----|----|------------|----|----|----|-----------|-----|-----|----|----|-----|----|----|----|----|----|----|----|----|----|
| 73   | Γ  |     |    |            |    | 1  |    |           |     |     |    |    |     |    |    |    |    |    |    |    | •  | ł  |
| J5   |    | •   | •  | •          | •  | •  | •  | •         | •   | •   | •  | •  | •   | •  | •  | •  | •  | •  | •  | •  | •  | •  |
| 5P32 | 4  | S   | 12 | 13         | 14 | 15 | 16 | 17        | 18  | 19  | 21 | 22 | 23  | 25 | 26 | 27 | 32 | 33 | 34 | 35 | 36 | 39 |
| IANO | D5 | 010 | D8 | <b>D</b> 9 | D6 | D7 | 00 | <b>D1</b> | 013 | 012 | A4 | A5 | 011 | D4 | D3 | D2 | AO | A1 | A2 | A3 | A6 | A7 |
| 8266 |    | 15  |    |            |    |    | 2  | 0         | 14  | 12  | 4  | S  | 13  |    | 16 |    |    |    |    |    | AO |    |

Figure 77 – Connessioni suggerite per lavorare con moduli MIC e schede ESP8266

### 29.5. Sketch di esempio

Posizione: IoTPSP Sample Sketches\IoTPSP MIC

Gli sketch di esempio leggono il valore analogico del MIC è lo visualizzano sul serial monitor.

Usare lo sketch IoTPSP\_MIC\_NANO per testare il MIC con schede ARDUINO NANO. Usare lo sketch IoTPSP\_MIC\_ESP32 per testare il MIC con schede with ESP32. Usare lo sketch IoTPSP\_MIC\_ESP8266 per testare il MIC con schede ESP8266.

### 29.6. Utilizzare i moduli MIC con pin GPIO differenti

Come mostrato in Figure 71, il segnale analogico dei moduli MIC è collegato al pin GPIO di default attraverso JP28.

Per disconnettere il segnale dal pin GPIO di default, APRIRE JP28 (incidere nel mezzo).

Tramite dei cavetti volanti, collegare il pin #2 dell'header J22 ad un altro pin GPIO secondo le proprie esigenze.

IOT PROTO SHIELD PLUS MANUALE UTENTE

Verificare le connessioni default della IOT PROTO SHIELD PLUB (vedere §8) per evitare conflitti con altre connessioni predefinite.

IOT PROTO SHIELD PLUS MANUALE UTENTE

# 30. SERVO

La IoT PROTO SHIELD PLUS è predisposta per lavorare con dei SERVI standard a 5V di larga diffusione sul mercato.

Per utilizzare un SERVO, è necessario saldare un pinstrip fila singola a 3 pin con passo standard 2,54mm (non fornito con la scheda) nella posizione J39.



Figure 78 - Sezione SERVO della IOT PROTO SHIELD PLUS



Figure 79 – Schema connessioni della sezione SERVO della IDT PROTO SHIELD PLUB

### 30.1. Connessioni default

|       | ESP32  | ARDUINO | ESP8266 |
|-------|--------|---------|---------|
| SERVO | GPIO04 | D5      | NC      |

NC = Non Connesso

L'alimentazione del SERVO è connessa di default a 5V.

Si noti che non ci sono connessioni predefinite per schede ESP8266. Per utilizzare un SERVO con schede ESP8266, è necessario eseguire i collegamenti con dei cavetti volanti.

### 30.2. Connessioni suggerite per lavorare con schede ESP8266

Lo sketch di esempio è fatto per lavorare con le seguenti connessioni: Pin GPIO2 (D4) dell'ESP8622 connesso al pin GPIO04 dell'ESP32 (SERVO).





### 30.3. Sketch di esempio

Posizione: IoTPSP Sample Sketches\IoTPSP SERVO

Gli sketch di esempio usano il POT per muovere il servo. Ruotare il POT per cambiare la posizione angolare del servo.

Vedere §11 (POT) per maggiori dettagli relativi all'utilizzo del POT sulla IOT PROTO SHIELD PLUS.

#### 30.3.1. IoTPSP\_SERVO\_NANO

Lo sketch IOTPSP SERVO NANO, fa uso della libraria Servo.

Assicuratevi di averla installata prima di usare lo sketch.

Assicurarsi inoltre si aver installato l'ultima versione della libreria: vecchie version della libreria non funzionano con schede Arduino NANO33 o EVERY.

Maggiori dettagli al link: <u>https://www.arduino.cc/reference/en/libraries/servo/</u>

| 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+I |  |  |  |  |  |  |
| This is an the sketch m                             | Serial Monitor        | Ctrl+Shift+M |  |  |  |  |  |  |

| 🐵 Library Manager  | ×  |
|--|----|
| Type All Topic All servo   | ^  |
| More info Version 0.0.13 V Install Servo   | 1  |
| by Michael Margolis, Arduino Version 1.1.7 INSTALLED<br>Allows Arduino boards to control a variety of servo motors. This library can control a great number of servos.<br>It makes careful use of timers: the library can control 12 servos using only 1 timer.<br>On the Arduino Due you can control up to 60 servos.<br><u>More info</u> |    |
| Accessories<br>by Thierry Paris - Locoduino<br>This is a library for Arduino to handle accessories like lights, motors. This library can handle coil motors, stepper motors,<br>lights, servos.  | -  |
| Clo  | se |

Utilizzare lo sketch <code>IoTPSP\_SERVO\_NANO</code> per testare un SERVO con schede ARDUINO NANO.

#### 30.3.2. IoTPSP\_SERVO\_ESP32

Lo sketch IOTPSP\_SERVO\_ESP32, fa uso della libraria *Esp32Servo*. Assicuratevi di averla installata prima di usare lo sketch. Maggiori dettagli al link: https://www.arduino.cc/reference/en/libraries/esp32servo/

| ioTPSP_SSD1306  | _DISPLAY_128x64_NANO   Arc   | duino 1.8.13   |   |   |   |        |
|---|--|--|---|---|---|--------|
|   | Auto Format<br>Archive Sketch<br>Fix Encoding & Reload   |  | Ctrl+T  |   |   |        |
| /***********<br>This is an<br>the sketch m  | Manage Libraries   |  | Ctrl+Shift+I<br>Ctrl+Shift+M                    | R   |   |        |
| <ul> <li>Library Manage</li> <li>Type All</li> </ul>  | v Topic All  | ✓ esp32servo   |   |   |   | ×      |
| <br>ESP32Servo<br>by Kevin Harrin<br>Allows ESP32 b<br>types of servos.<br>It makes use o<br>No attempt has<br><u>More info</u> | gton,John K. Bennett<br>oards to control servo, ton<br>f the ESP32 PWM timers: ti<br>been made to support mu | ne and analogWrit<br>he library can coni<br>ultiple servos per ( | te motors using<br>trol up to 16 se<br>channel. | <b>g Arduino semantics.</b><br>ervos on individual ch | This library can control a m<br>annels<br>Version 0.9.0 V | iany ^ |
| ESP32Servo360<br>by Sébastien M.<br>Initial developm<br>More info   | atos<br>nent release Control Parall  | lax Feedback 360   | ° High Speed S                                  | iervos with a ESP32 d                                 | lev-board.  |        |
|   |  |  |   |   |   | Close  |

Utilizzare lo sketch IoTPSP\_SERVO\_ESP32 per testare un SERVO con schede ESP32.

#### 30.3.3. IOTPSP SERVO ESP8266

Lo sketch IOTPSP\_SERVO\_ESP8266, fa uso della libraria Servo (vedi § 30.3.1). Assicuratevi di averla installata prima di usare lo sketch. Il segnale deve essere connesso secondo le indicazioni mostrate in Figure 80.

Utilizzare lo sketch IoTPSP\_SERVO\_ESP8266 per testare un SERVO con schede ESP8266.

### 30.4. Utilizzare un SERVO con pin GPIO differenti

Come mostrato in Figure 79, il segnale del SERVO è collegato al pin GPIO di default attraverso JP53. Per disconnettere il segnale dal pin GPIO di default, APRIRE JP53 (incidere nel mezzo).

Tramite dei cavetti volanti, collegare il pin #1 dell'header J30 ad un altro pin GPIO secondo le proprie esigenze.

Verificare le connessioni default della IOT PROTO SHIELD PLUB (vedere §8) per evitare conflitti con altre connessioni predefinite.

### 30.5. Powering the SERVO with a different power supply

Come mostrato in Figure 79, l'alimentazione *Vservo* è collegata di default a 5V attraverso JP54. La prima cosa da fare è disconnettere *Vservo* dai 5V (incidere JP54 tra i pad 1 e 2).

Assicurarsi che la connessione tra i PAD 1-2 di JP54 sia APERTO per evitare danni alla IoT PROTO SHIELD PLUS.

A questo punto è possibile collegare mediante JP52 e JP54 (vedi gure 78).

In alternativa è possibile utilizzare il pin #2 dell'header J30 per collegare un'alimentazione esterna, nel caso ricodarsi che le masse (GND) devono essere in comune.

# 31. HC-SR04 (sensore di distanza)

La **IDT PROTO SHIELD PLUS** è predisposta per lavorare con moduli a 4 pin HC-SR04 di larga diffusione sul mercato (vedi Figure 81).

L'alimentazione del modulo HC-SR04 deve essere 5V.

Il segnale TRIGGER (HC\_SR\_T) lavora sia a 5V che 3V3, quindi non ha importanza se stiate lavorando con una scheda a 5V (es. NANO o EVERY) o a 3V3 (NANO33 o ESP).

Essendo l'alimentazione del modulo 5V, il segnale ECHO sarà anch'esso un segnale a 5V, la IDT PROTO SHIELD PLUS è provvista di un circuito di limitazione (clipping) che permette ai moduli HC-SR04 di lavorare con schede 3V3 (es. ESP, NANO33, etc.)

Per lavorare con un modulo HC-SR04, inserirlo nell'header J14 come mostrato di seguito.



Figure 81 – Modulo HC-SR04 di larga diffusione sul mercato



Figure 82 – Sezione HC-SR04 della IDT PROTO SHIELD PLUS



Figure 83 – Schema collegamenti della sezione HC-SR04 della IDT PROTO SHIELD PLUS

 $10_{2}$ 



Figure 84 – Un modulo HC-SR04 inserito nell'header J14 della IOT PROTO SHIELD PLUS

#### 31.1. Connessioni di default

|         | ESP32  | ARDUINO | ESP8266 |
|---------|--------|---------|---------|
| HC_SR_T | GPIO14 | D6      | NC      |
| HC_SR_E | GPIO15 | D7      | NC      |

*NC* = Non Connesso

Si noti che non ci sono connessioni predefinite per schede ESP8266. Per utilizzare moduli HC-SR04 con schede ESP8266, è necessario eseguire i collegamenti con dei cavetti volanti.

### 31.2. Connessioni suggerite per lavorare con moduli HC-SR04 e schede ESP8266

Gli sketch di esempio sono fatti per lavorare con le seguenti connessioni: Pin GPIO4 (D2) dell'ESP8622 connesso al pin GPIO14 dell'ESP32 (HC\_SR\_T). Pin GPIO5 (D1) dell'ESP8622 connesso al pin GPIO15 dell'ESP32 (HC\_SR\_E).



Figure 85 – Connessioni suggerite per lavorare con moduli HC-SR04 e schede ESP8266

### **31.3.** Sketch di esempio

Posizione: IoTPSP\_Sample\_Sketches\IoTPSP\_HC\_SR04

Utilizzare lo sketch IoTPSP\_HC\_SR04\_NANO per testare moduli HC-SR04 con schede ARDUINO NANO. Utilizzare lo sketch IoTPSP\_HC\_SR04\_ESP32 per testare moduli HC-SR04 con schede ESP32. Utilizzare lo sketch IoTPSP\_HC\_SR04\_ESP8266 per testare moduli HC-SR04 con schede ESP8266.

Open the serial monitor to see the distance measured by the SR-HC04 sensor. Aprire il Serial Monitor per visualizzare la distanza misurata dal sensore SR-HC04.

Maggiori dettagli al link: <u>https://create.arduino.cc/projecthub/Isaac100/getting-started-with-the-hc-sr04-ultrasonic-sensor-036380</u>

### 31.4. Utilizzare moduli HC-SR04 con pin GPIO differenti

Come mostrato in Figure 83, il segnali HC\_ST\_T and HC\_SR\_E sono connessi ai pin GPIO di default attraverso JP39 e JP40.

Per disconnettere i segnali dai pin GPIO di default, APRIRE JP39 e JP40 (incidere nel mezzo).

Tramite dei cavetti volanti, collegare l'header J17 ad altri pin GPIO secondo le proprie esigenze.

Verificare le connessioni default della IOT PROTO SHIELD PLUB (vedere §8) per evitare conflitti con altre connessioni predefinite.

## 32. MOTORE PASSO-PASSO

La **IOT PROTO SHIELD PLUS** è predisposta per lavorare con driver per motori passo-passo che hanno un fattore di forma compatibile con i famosi moduli DRV8825.

Questi moduli sono usati per pilotare motori passo-passo a 2 fasi e 4 poli (A, /A, B, /B). Il digramma delle connessioni varia leggermente a seconda di quale driver si sta utilizzando (si veda di seguito), tuttavia ci sono alcuni collegamenti necessari che si applicano a tutti i modelli di driver:

- Alimentazione Motore
- Alimentazione della parte Logica
- Segnale di Sleep (e Reset)
- Segnale di Step
- Segnale di Direzione

Come mostrato in Figure 86 e §32.4.1, la **IDT PROTO SHIELD PLUS** è predisposta per lavorare con un driver DRV8825 senza alcuna connessione (o disconnessione) aggiuntiva.

Si consideri che *Vmot* è disconnesso di default: spetta a voi alimentare (J12) il motore con la tensione corretta considerando le connessioni mostrate in Figure 87.

Si consiglia vivamente la lettura completa di questo capitolo (§32) prima di utilizzare un motore passopasso: è bene conoscere tutti gli aspetti e le opzioni predisposte sulla IOT PROTO SHIELD PLUS, in modo da poter effettuare le scelte necessarie al corretto utilizzo del vostro motore passo-passo.



Figure 86 – Schema connessioni della sezione Motore passo-passo della Iot Proto Shield Plus



Figure 87 – Schema connessioni relative alla tensione di alimentazione Vmot della IoT PROTO SHIELO PLUS

#### Il socket BRD1 (MT3608 step-up-Module) è una predisposizione per possibili applicazioni future, non considerarlo al momento.

| USB<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+2506<br>UN+250 | OWNE CTOP         Value         Owne         Owne | 30         Image: Point - Vouril-         Ima |  |
|---|---|--|--|
| 277<br>277<br>277<br>277<br>277<br>277<br>277<br>277<br>277<br>277  |   |  |  |
| CAN<br>West<br>D 31   |   |  |  |

Figure 88 – Socket BRD2 della IOT PROTO SHIELD PLUS



Figure 89 – Connettore a vite J15 della IDT PROTO SHIELD PLUS

### 32.1. Connessioni di default

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

NC = Non Connesso

Si noti che non ci sono connessioni predefinite peri il segnale MOT\_DIR per schede ESP8266. Per utilizzare un motore passo-passo con schede ESP8266, è necessario collegare il segnale MOT\_DIR con dei cavetti volanti.

### 32.2. Connessioni suggerite per lavorare con schede ESP8266

Lo sketch di esempio è fatto per lavorare con le seguenti connessioni: Pin GPIO14 (D5) dell'ESP8622 connesso al pin GPIO25 dell'ESP32 (MOT\_DIR).



Figure 90 – Connessioni suggerite per lavorare con motori stepper e schede ESP8266

### 32.3. Sketch di esempio

Posizione: IoTPSP Sample Sketches\IoTPSP STEPPER MOTOR

Lo sketch fa eseguire una rotazione del motore in senso orario ed una rotazione in senso antiorario continuando a ciclo continuo.

È necessario impostare il numero di STEP per GIRO del motore in utilizzo, affinché la rotazione sia di 360°.

#### 32.3.1. IoTPSP\_STEPPER\_MOTOR\_NANO

Lo sketch IOTPSP\_STEPPER\_MOTOR\_NANO fa uso della libreria StepperDriver.

Assicuratevi di averla installata prima di usare lo sketch.

Maggiori dettagli al link: https://www.arduino.cc/reference/en/libraries/stepperdriver/

| IoTPSP_SSD1306   | DISPLAY_128x64_NANO   Ardu  | ino 1.8.13  |
|--|---|---|
|  | Auto Format<br>Archive Sketch   | Ctrl+T  |
| IoTPSP_SSD1<br>/********   | Fix Encoding & Reload<br>Manage Libraries   | Ctrl+Shift+I  |
| This is an the sketch m  | Serial Monitor  | Ctrl+Shift+M  |
| C Library Manager<br>Type All<br>HighPowerSteppert<br>by Pololu<br>High-Power Stepper<br>More info                             | Topic All Ste<br>Driver<br>er Motor Driver library for Arduin<br>r Motor Drivers.                                   | perDriver   |
| StepperDriver<br>by Laurentiu Bade<br>A4988, DRV8825<br>the ones from Polo<br>DRV8834, DRV888<br>More info<br>Select version V | a Version 1.3.1 INSTALLED<br>and generic two-pin stepper moto<br>lu. Microstepping is supported. An<br>0.<br>Instal | <b>r driver library.</b> Control steppers via a driver board providing STEP+DIR like<br>celeration is supported. Supported drivers are A4988, DRV8824, DRV8825, |
|  |   | Close   |

#### 32.3.2. IOTPSP STEPPER MOTOR ESP32

Lo sketch IOTPSP STEPPER MOTOR ESP32 usa la libreria ESP-FlexyStepper. Assicuratevi di averla installata prima di usare lo sketch.

Maggiori dettagli al link: https://www.arduino.cc/reference/en/libraries/esp-flexystepper/

| - 00 -                  |                                  |              |
|-------------------------|----------------------------------|--------------|
| IoTPSP_SSD13            | 06_DISPLAY_128x64_NANO   Arduind | 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 |
|                         |                                  |              |
|                         |                                  |              |

| Colorary Manager   | ×          |
|--|------------|
| Type All v Topic All SP-flexy  |            |
| ESP-FlexyStepper         by Paul Kerspe Version 1.4.3 INSTALLED         This library is used to control one or more stepper motors from an ESP32 device This library is used to control one or more stepper motors from an ESP32 device. It is based on the FlexyStepper library by S.Reifels but provides some additional functionality         More info         Version 1.4.2       Instri- |            |
|  |            |
|  | ∨<br>Close |

#### 32.3.3. IOTPSP STEPPER MOTOR ESP8266

Lo sketch IOTPSP STEPPER MOTOR ESP8266 è basato sull'esempio "Connecting all control pins from DRV8825 to NodeMCU" descritto al link:

https://hackaday.io/project/160569-nodemcu-and-drv8825

#### 32.4. Scelta del modulo driver

Come accennato in precedenza, ci sono diversi driver per motori passo-passo, ognuno ha i suoi pro e contro.

# Indipendentemente dal modello di driver, la prima cosa da fare è impostare la corrente per fase in base al motore che si sta utilizzando.

La procedura è descritta al link: https://www.youtube.com/watch?v=89BHS9hfSUk

#### 32.4.1. Modulo DRV8825

Questo modulo è basato sul chip DRV8825.

Si consiglia di leggere il datasheet (<u>https://www.ti.com/lit/ds/symlink/drv8825.pdf</u>) prima di utilizzare questo modulo.

Una spiegazione dettagliata di questi moduli (ed in generale moduli basati sul chip DRV8825) è disponibile al link: <u>https://www.pololu.com/product/2133</u>

- PRO
  - o **Economici**
  - Di largo utilizzo
  - Facili da reperire
  - La IDT PROTO SHIELD PLUS è predisposta per lavorare con questi moduli senza che sia richiesta nessuna modifica ai jumper di configurazione (si veda Figure 86 e Figure 91)
- CONTRO
  - Richiede una tensione di alimentazione del motore compresa tra 8.2V e 45V, quindi è necessaria una alimentazione esterna per utilizzare il motore



Figure 91 – Connessioni minime per pilotare motori passo-passo con moduli DRV8825 (full-step mode)

Per lavorare con moduli DRV8825 sulla IOT PROTO SHIELD PLUS:

- Come mostrato in Figure 95 e Figure 86, JP27 e JP33 devono essere CHIUSI (configurazione di default), così da connettere Vbrd della IDT PROTO SHIELD PLUS ai pin di RESET e SLEEP del modulo DRV8834
- Inserire il modulo nel socket BRD2 facendo attenzione all'orientamento dei pin (vedere Figure 92)
- Connettere i poli A, /A, B, /B del motore secondo le specifiche di collegamento e Figure 89
- Connettere l'alimentazione del motore secondo le specifiche di funzionamento e Figure 89



Figure 92 – Un modulo DRV8825inserito nel socket BRD2

#### 32.4.2. Moduli TMC2xxx

Questi moduli sono basati sui chip TMC21xx to TMC26xx. Si raccomanda di leggere il datasheet (<u>https://www.trinamic.com</u>) prima di utilizzare questi moduli.

- PRO
  - o Noiseless current control (rotazione motori con estrema silenziosità)
  - Sempre più utilizzati e quindi destinati ad essere di largo utilizzo (almeno al momento di stesura di questo manuale)
  - Facilmente reperibilil sul mercato
  - Sono richieste minime modifiche alle configurazione di default della IDT PRDTD
     SHIELD PLUS per l'utilizzo di questi moduli (si veda di seguito)
  - o Necessita di una tensione da 3V a 5V per l'alimentazione della parte logica
  - Lavora con tensioni di alimentazione del motore da 5V a 36V (che significa che con correnti di fase del motore inferiori a 0.5A è possibile utilizzare i 5V forniti dalle prese USB del PC per poter alimentare il motore)
- CONTRO
  - Leggermente più costosi dei DRV8825

Considerando i PRO ei CONTRO sopra menzionati, i moduli basati su chip TMC sono la scelta più versatile per sperimentare con motori passo-passo.



Per lavorare con moduli TMC2xxx sulla IDT PROTO SHIELD PLUS:

- Come mostrato in Figure 95 e Figure 86, JP29 deve essere CHIUSO così da connettere il pin VIO del modulo TMC2xxx a Vbrd della IDT PROTO SHIELD PLUS
- Inserire il modulo TMC2xxx nel socket BRD2 facendo attenzione all'orientamento dei pin (vedere Figure 94)
- Connettere i poli A, /A, B, /B del motore secondo le specifiche di collegamento e Figure 89
  - Alimentare il motore, secondo le specifiche di funzionamento, Figure 87 e Figure 89
    - Per utilizzare i 5V delle prese USB del PC, è necessario CHIUDERE JP31 (con una goccia di stagno)
- Questi moduli richiedono che il pin EN sia BASSO per abilitare la rotazione (collegare J13 Pin#5 a GND)



Figure 94 – Un modulo TMC2xxx inserito nel socket BRD2

#### 32.4.3. Modulo DRV8834

Questi moduli sono basati sul chip DRV8834.

Si raccomanda di leggere il datasheet (<u>https://www.ti.com/lit/ds/symlink/drv8834.pdf</u>) prima di utilizzare questi moduli.

Una spiegazione dettagliata di questi moduli (ed in generale moduli basati sul chip DRV8834) è disponibile al link: <u>https://www.pololu.com/product/2134</u>

• PRO

•

- Lavora con tensioni di alimentazione del motore da 2.5V a 10.8V (che significa che con correnti di fase del motore inferiori a 0.5A è possibile utilizzare i 5V forniti dalle prese USB del PC per poter alimentare il motore)
- Sono richieste minime modifiche alle configurazione di default della IDT PRDTD
   SHIELD PLUS per l'utilizzo di questi moduli (si veda di seguito)
- CONTRO
  - o Decisamente più costosi se comparati ad altri driver per motori passo-passo
  - o Difficilmente reperibili sul mercato
  - Non adatti a pilotare motori con tensioni di alimentazione maggiori di 10.2V (comunque capaci di pilotare motori NEMA17 o simili)



Figure 95 – Connessioni minime per pilotare motori passo-passo con moduli DRV8834 (full-step mode)

Per lavorare con moduli DRV8834 sulla IDT PROTO SHIELD PLUS:

- Come mostrato in Figure 95 and Figure 86, JP27 deve essere CHIUSO così da connettere il pin *SLEEP* del modulo DRV8834 a *Vbrd* della Iot Proto Shield Plus
- Come mostrato in Figure 95 and Figure 86, JP33 deve essere APERTO in modo da disconnettere il pin *VREF* del modulo DRV8834 a *Vbrd* della IDT PROTO SHIELO PLUS
- Inserire il modulo DRV8834 nel socket BRD2, facendo attenzione all'orientamento dei pin (vedere Figure 96)
- Connettere i poli A, /A, B, /B del motore secondo le specifiche di collegamento e Figure 89
- Alimentare il motore, secondo le specifiche di funzionamento, Figure 87 e Figure 89
  - Per utilizzare i 5V delle prese USB del PC, è necessario CHIUDERE JP31 (con una goccia di stagno)



Figure 96 – Un modulo DRV8834 inserito nel socket BRD2

#### 32.4.4. Modulo A4988

Questo modulo è basato sul chip A4988.

Si consiglia di leggere il datasheet (<u>https://www.pololu.com/file/0J450/A4988.pdf</u>) prima di utilizzare questo modulo.

Una spiegazione dettagliata di questi moduli (ed in generale moduli basati sul chip A4988) è disponibile al link: <u>https://www.pololu.com/product/1182</u>

# www.GtronicsShop.com www.Gtronics.NET

- PRO
  - Ragionevolmente economici
  - o DI largo utilizzo
  - Facilmente reperibili sul mercato
  - Sono richieste minime modifiche alle configurazione di default della IDT PRDTD
     SHIELD PLUS per l'utilizzo di questi moduli (si veda di seguito)
- CONTRO
  - Richiede una tensione di alimentazione del motore compresa tra 8V e 35V, quindi è necessaria una alimentazione esterna per utilizzare il motore





Figure 97 – Connessioni minime per pilotare motori passo-passo con moduli A4988 (full-step mode)

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

- Come mostrato in Figure 95 e Figure 86, JP27 deve essere APERTO, così da disconnettere *Vbrd* della IDT PRDTD SHIELD PLUS dai pin di RESET e SLEEP del modulo A4988
- Come mostrato in Figure 95 e Figure 86, JP33 deve essere CHIUSO, così da connettere i pin RESET e SLEEP del modulo A4988 tra loro
- Come mostrato in Figure 95 e Figure 86, JP29 deve essere CHIUSO, così da connettere Vbrd della IDT PROTO SHIELD PLUS al pin VDD del modulo A4988



Figure 98 – Un modulo A4988 inserito nel socket BRD2

### 32.5. Scelta e connessione del motore passo-passo

A seconda dalla vostra applicazione, sceglierete un motore che meglio si adatta alle vostre esigenze. In base alle specifiche del motore, è possibile scegliere il modulo driver più adatto.

Di seguito sono riportati due esempi di motori passo-passo:

- Motore passo-passo NEMA17 di largo utilizzo sul mercato (vedi Figure 99)
- micro motore passo-passo (vedi Figure 100).



Figure 99 – motore passo-passo NEMA 17



Figure 100 – Micro motore passo-passo

### 32.6. Utilizzare motori passo-passo con pin GPIO differenti e funzioni aggiuntive

Come mostrato in Figure 86, i segnali MOT\_DIR e MOT\_STP sono collegati al pin GPIO di default attraverso JP37 e JP38.

Per disconnettere i segnali MOT\_DIR e MOT\_STP dai pin GPIO di default, APRIRE JP37 eJP38 (incidere nel mezzo).

Tramite dei cavetti volanti, collegare i pin #6 e #7 dell'header J13 ad un altro pin GPIO secondo le proprie esigenze.

Come mostrato in Figure 91, Figure 95 e Figure 97, alcuni pin (come ad esempio quelli per la regolazione dei microsteps) non sono connessi di default, è possibile accedere alle funzionalità aggiuntive fornite da questi pin attraverso l'header J13, fare rifermento a Figure 86.

Verificare le connessioni default della IoT Proto Shield Plus (vedere §8) per evitare conflitti con altre connessioni predefinite.

# **33. LEVEL SHIFTER**

Il level shifter intergrato è una copia del modulo level shifter di larga diffusione.

Il level shifter fornisce connessioni HV (HighVoltage) e LV (LowVoltage) (vedere la Figure 102). Il level shifter converte un segnale di tensione HV in un segnale di tensione LV e, viceversa, converte un segnale di tensione LV in un segnale di tensione HV.

Di default, HV è connesso ai 5V (tramite JP41) e LV è connesso ai 3V3 (tramite JP42).





Figure 101 – LA sezione Level Shifter della IOT PROTO SHIELD PLUS



Figure 102 – Schema connessioni della sezione Level Shifter della IoT PROTO SHIELD PLUS

### 33.1. Utilizzare il level shifter integrato con tensioni HV e LV differenti

Per utilizzare il level shifter integrato con tensioni HV differenti, è necessario scollegare il pad #1 di JP41 dal pad #2 (incedere tra i due pad) e collegare il pad #2 di JP41 al pad #3 (con una piccola goccia di stagno). Collegare poi pin #3 dell'header J16 la tensione HV desiderata.



Figure 103 – JP41 e pin#3 dell'header J16 della sezione Level Shifter della IoT PROTO SHIELD PLUS

# www.GtronicsShop.com www.Gtronics.NET

IOT Proto Shield Plus Manuale Utente

Per utilizzare il level shifter integrato con tensioni LV differenti, è necessario scollegare il pad #1 di JP42 dal pad #2 (incedere tra i due pad) e collegare il pad #2 di JP42 al pad #3 (con una piccola goccia di stagno). Collegare poi pin #3 dell'header J18 la tensione LV desiderata.



Figure 104 – JP42 e pin#3 dell'header J18 della sezione Level Shifter della IoT PROTO SHIELD PLUS

# 34. Modulo WiFi ESP01

Se si desidera utilizzare un modulo ESP01WiFi (antenato dei moduli ESP8266) è possibile saldare un header 2x4 in posizione BRD6 e saldare header 1x8 in posizione J42 (vedi Figure 105). J42 consente di accedere ai pin ESP01 più facilmente rispetto al suo footprint nativo 2x4.





Figure 105 – La sezione ESP01 della IOT PROTO SHIELD PLUS



Figure 106 – Schema connessioni della sezione ESP01 della IOT PROTO SHIELD PLUS

#### IOT PROTO SHIELD PLUS MANUALE UTENTE

### **35. ALIMENTAZIONE ESTERNA**

#### DISCLAIMER

Questa sezione è dedicata ad utenti esperti, inoltre, questa sezione non è popolata sulla scheda. Per utilizzare correttamente un alimentazione esterna abbinata ad un modulo DC-DC step-down LM2596, fare riferimento agli schemi della **Iот Ркото Sнієсо Рсиз** (vedi §36) e alle specifiche tecniche dei dispositivi in uso.

La scelta di un regolatore di tensione errato o errori nella connessione dell'alimentazione Vin (vedere gli schemi al §36) potrebbe danneggiare la scheda ESP o NANO in uso.

Per collegare il modulo LM2596 sono necessari quattro connettori a 2 pin (non inclusi nella IoT PROTO SHIELD PLUS) da saldare ai pad Vin+, Vin-, Vout+ e Vout- della IoT PROTO SHIELD PLUS. L'alimentazione esterna può essere fornita tramite connettore con morsetti a vite JP2 (vedi Figure 108 e Figure 110).

Se non si dispone di un connettore a 2 terminali a vite, è possibile utilizzare i due pad della scheda e saldare due fili direttamente su di essi.

Il connettore con terminali a vite non è fornito la IDT PROTO SHIELD PLUB e deve essere acquistato e saldato autonomamente.

L'alimentazione esterna si collega a IN+ (Vin+ sulla serigrafia) del modulo LM2596 tramite JP57, che deve essere CHIUSO (con una piccola goccia di stagno).

Se si desidera aggiungere una protezione per evitare l'inversione di polarità, è possibile aggiungere D4 (si suggerisce 1N5819, vedere Figure 110) ma in questo caso JP57 deve essere APERTO.

La tensione regolata OUT+ (Vout+ sulla serigrafia) del modulo LM2596 si collega a Vin della IDT PROTO SHIELD PLUS.





Figure 107 – UN modulo DC-DC LM2596 di largo utilizzo



Figure 108 – La sezione relativa all'alimentazione esterna della IOT PROTO SHIELD PLUS



Figure 109 – Un moduloDC-DC LM2596 inserito nella IoT PROTO SHIELD PLUS



Figure 110 – Schema connessioni della sezione relativa all'alimentazione esterna della IoT PROTO SHIELD PLUS

### 35.1. Battery Shield

In alternativa al modulo LM2596, è possibile utilizzare un modulo battery shield di larga diffusione (vedere Figure 111).

Il modulo battery shield viene utilizzato per incrementare i 3.7Vdc di una batteria Lipo ad un elemento, fino a 5V così da poter essere connesso al pin Vin delle schede ESP32 o Vin delle schede NANO (vedi Figure 114). Collegare la batteria Lipo al modulo battery shield tramite il suo connettore JST.

Per collegare il modulo battery shield, sono necessari due connettori a 8 pin (non inclusi nella IoT Proto SHIELD PLUS) che vanno saldati ai pad mostrati nella Figure 112 (BRD5).

Collegare poi il modulo battery shield secondo l'orientamento mostrato in Figure 113.



Figure 111 – Tipico modulo battery shield di largo utilizzo



Figure 112 – Il socket BRD5 per collegare il modulo battery shield sulla IDT PROTO SHIELO PLUS



Figure 113 – Modulo battery shield inserito nel socket BRD5 della IOT PROTO SHIELD PLUS

| Battery_Sh        | BRD5<br>ield_8266_m | ini                            |
|-------------------|---------------------|--------------------------------|
| ×1<br>×2          | 16<br>15            | ×<br>×                         |
| ×-5<br>×-4<br>×-5 | 13                  | ×<br>×                         |
| ×6<br>×7<br>×8    | GND 10<br>5V 9      | ×<br>↓<br>↓<br>GND<br>↓<br>Vin |

Figure 114 – Schema connessioni della sezione battery shield della IoT PROTO SHIELD PLUS

IOT PROTO SHIELD PLUS MANUALE UTENTE Rev.1.07.i





IoTPSP\_User\_Manual\_IT\_r1.0.docx
## www.GtronicsShop.com www.Gtronics.NET



Rev.1.07.i



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





Rev.1.07.i



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



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

111/111