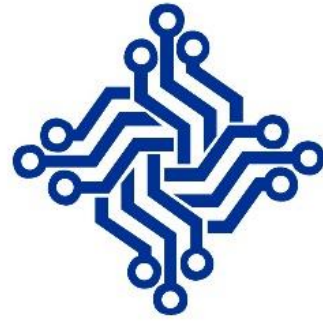
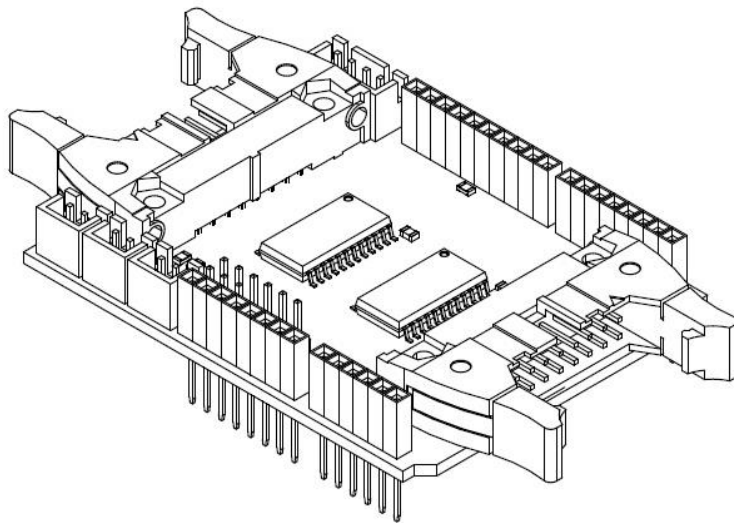


User Manual



ERIDEVICES



***Arduwire UNO Shield,
Universal Continuity Cable Tester***

MODEL ARDUWIRE UNO

USER MANUAL

Index

1.1 Introduction.

1.2 Features.

1.3 Overview Hardware.

1.4 Accessories and options.

1.5 Installation of Library.

1.6 Overview Library.

1.7 Code for arduino.

Instruction Manual

For the
Arduwire UNO
Universal Cable Tester

by

Eridevices

1.1 Introduction

The **Arduwire UNO** is a portable shield and stand-alone cable / harness tester for cables of up to 32 test points. The shield detects opens, shorts, and miswires within one second by comparing test results with a golden file or from a sample cable. Upon error detection, an error can be displayed on the LCD display and error of the LED indicator and speaker.

The shield is only compatible with **arduino UNO**, and it is possible to perform test all kind of cables IDC, dsub, USB, RJ45.

1.2 Features:

- Easy to use.
- Expandable 64 TP.
- Learn Function.
- Buzzer output.
- Library for arduino.
- Calculate CRC8.
- LCD I2C display compatible.
- Compatible with Arduino UNO.
- Universal cable tester.
- Pass/Fail LED Indicator.

1.3 Overview Hardware

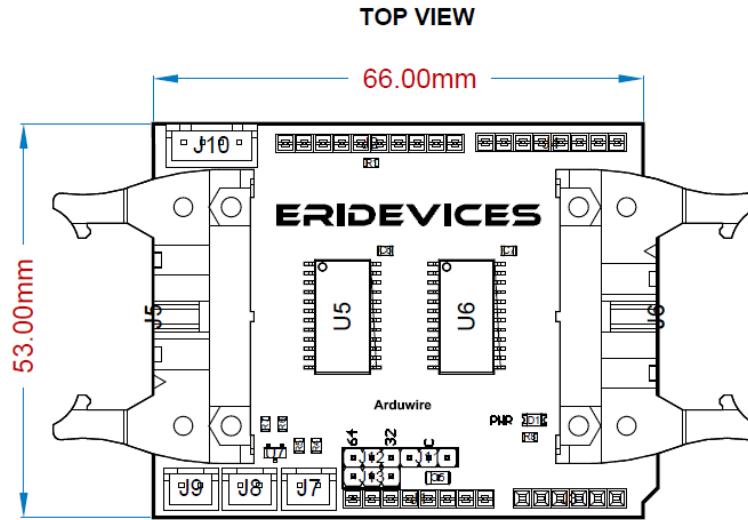


Image 1. – Arduwire UNO, Dimensions.

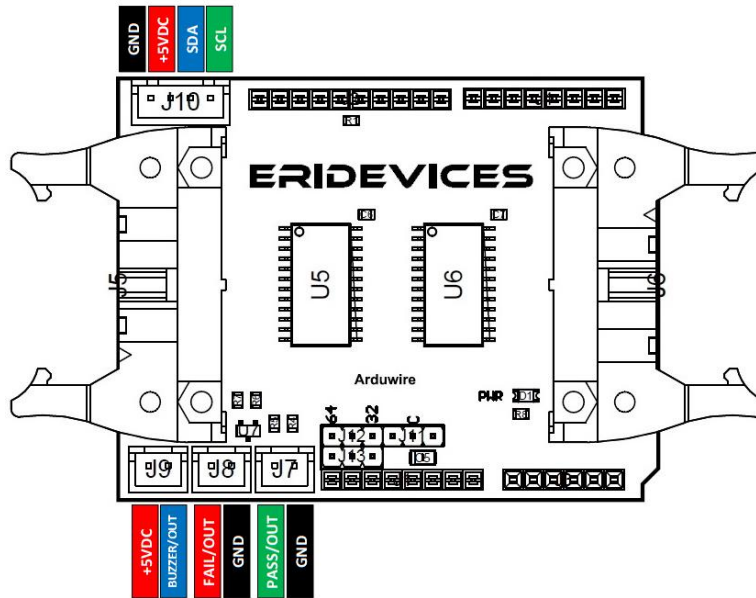


Image 2. - Arduwire UNO Pinout.

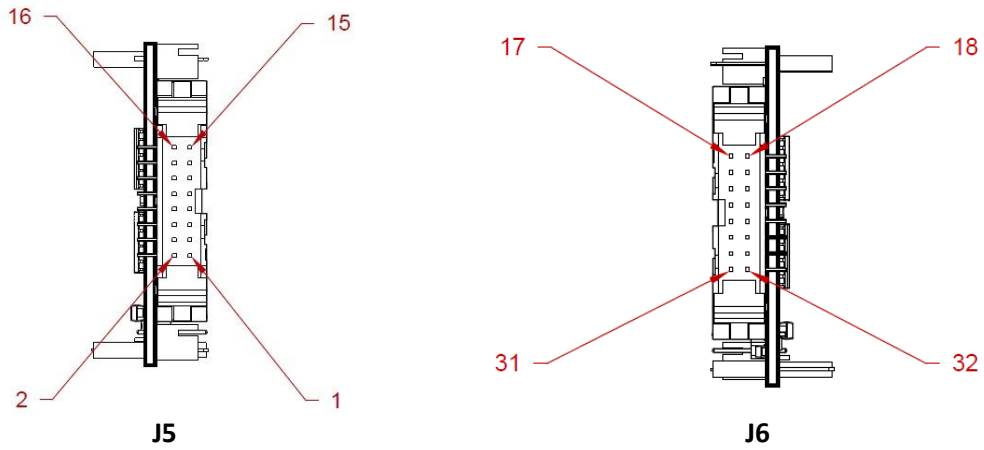


Image 3. - Outputs & Inputs.

Specs

Designator	Mode	Description
J5	OUT/IN	Outputs test points (1-16).
J6	OUT/IN	Outputs test points (17- 32).
J7	OUT	Green led "PASS".
J8	OUT	Red led "FAIL".
J9	OUT	Buzzer.
J10	OUT	LCD I2C / BUS.
J1	-	Selector expansion.
J12	-	Selector expansion.
J13	-	Selector expansion.

1.4 Accessories and Options

The following accessories are provided with each Arduwire UNO shield:

- Arduwire UNO shield.
- User Manual (please visit the web page: eridevices.com).
- Set of cables with led's (optional).
- Buzzer (optional).

Options:

Expandable of 64 TP, is necessary connect another Arduwire UNO mode "stackable mode".

Arduwire UNO selector.

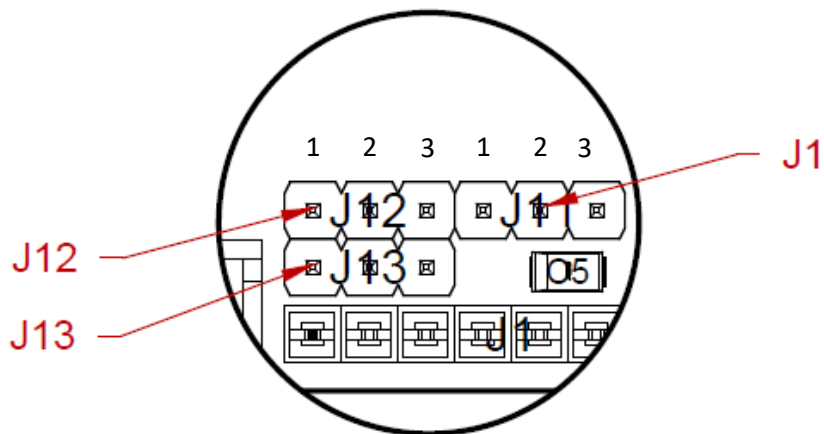


Image 2. – Selector

Set the Arduwire UNO for mode 32 TP:

Designator	Positions		
	1	2	3
J1	X	X	Expansion >>
J12		X	X
J13		X	X

Set the Arduwire UNO for mode 64 TP (Second Board):

Designator	Positions		
	1	2	3
J1		Expansion>>	
J12	X	X	
J13	X	X	

X = Jumper, Expansion >> = Connection J1-3 board 1 to J1-2 board 2.

1.5 Installation of Library

Installation of the library (*Arduwire UNO*).

1.- please visit the web page:

[GitHub - Erizm/ArduwireUNO-REV2: Arduwire UNO shield for arduino UNO](https://github.com/Erizm/ArduwireUNO-REV2)

2.- Download the zip file and unzipped in this root:

C:\Program Files\arduino-1.8.10\libraries

3.- Please restart the IDE of arduino.

4.- Now the library it is ready for use. Opening the example in the IDE of arduino:

File/Examples/arduwire/blink_arduwire

5.- Attach the Arduwire UNO shield in the arduino, the accessories showed in the image 3 is only for reference.

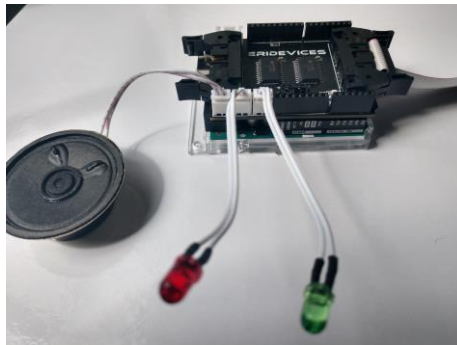


Image 3.- Arduwire UNO shield.

6.- Upload the sketch “blink_arduwire”.

7.- The blinking sketch it is only for verify the correct install of the library.

1.6 Overview Library

```
//Import Library
#include <arduwire.h>

//Create the object parameter maximum number of the test points.
arduwire ArduwireUNO(32);

//Initializing arduwire
ArduwireUNO.init();

//Function return boolean value.
ArduwireUNO.learn();

//Function return boolean value.
ArduwireUNO.test();

//Turn on pass led
ArduwireUNO.led_pass_on();

//Turn off pass led
ArduwireUNO.led_pass_off();

//Turn on fail led
ArduwireUNO.led_fail_on();

//Turn off fail led
ArduwireUNO.led_fail_off();

//Scanner option, return the test point
ArduwireUNO.scanner();

//Get the number of connections learned, return uint8_t
ArduwireUNO.get_connections_learned();
```

1.7 Code for arduino

```
/*
   THIS EXAMPLE IT IS COMPATIBLE WITH THE SHIELD ARDUWIRE UNO
   DEVELOPMENT BY ERIDEVICES MAR/07/20
   REVISED NOV/19/2020
*/

//IMPORT THE LIBRARY arduwire.h
#include <arduwire.h>

//CREATE THE OBJECT(MAX TEST POINTS)
arduwire ArduwireUNO(32);

void setup() {
    //INITIAL THE SERIAL PORT TO 9600 BAUD RATE
    Serial.begin(9600);
    //INITIAL THE ARDUWIRE INSTANCE
    ArduwireUNO.init();

    Serial.println(F("Verifying Connections.....\r"));

do{
    //THIS FUNCTION VERIFY ANY CONNECTION
    ArduwireUNO.learn();

}while(ArduwireUNO.get_learned_cable()!=true);

//NOW IT IS FOUNDED THE CONNECTIONS
Serial.print(F("Number of Connections Founded: "));
//THIS FUNCTION RETURN THE NUMBER OF CONNECTIONS FOUNDED
Serial.println(ArduwireUNO.get_connections_learned());
//SMALL ALARM BEEP
ArduwireUNO.beep_probe();
//VARIABLE COUNTER
uint8_t counter;
Serial.println(F("WIRELIST LEARNED:"));
for(uint8_t i = 0; i < ArduwireUNO.get_connections_learned(); i++){
    //INCREMENT
    counter++;
    Serial.print(counter);
    Serial.print(F(" "));
    Serial.print(ArduwireUNO.get_pina_learned(i));
    Serial.print(F(","));
    Serial.println(ArduwireUNO.get_pinb_learned(i));
}

}

void loop() {
    // put your main code here, to run repeatedly:

}
```