

```
//CHOOSE VERSION of tests
//Warning! clones with CH340 not working correctly
#define MICRO //Arduino Pro Micro
```

```
/*
```

```
Wiring test for Arduino UNO or Pro Micro
https://github.com/Twilight-Logic/AR488/
```

```
Warning! Not working with CH340 clones
```

```
Read instruction on serial monitor, baudrate 9200 (line #define BAUDRATE ...)
```

ARDUINO UNO

Arduino GPIB

pin	pin	function
-----	-----	----------

D2	10	SRQ
----	----	-----

D3	17	REN
----	----	-----

D7	11	ATN
----	----	-----

D8	9	IFC
----	---	-----

D9	8	NDAC
----	---	------

D10	7	NRFD
-----	---	------

D11	6	DAV
-----	---	-----

D12	5	EOI
-----	---	-----

A0	1	DIO1
----	---	------

A1	2	DIO2
----	---	------

A2	3	DIO3
----	---	------

A3	4	DIO4
----	---	------

A4	13	DIO5
----	----	------

A5	14	DIO6
----	----	------

D4	15	DIO7
----	----	------

D5	16	DIO8
----	----	------

GND	12	Shield
-----	----	--------

GND	18,19,20,21,22,23	GND
-----	-------------------	-----

ARDUINO MICRO PRO

Arduino GPIB

pin	pin	function
-----	-----	----------

A2	7	NRFD
----	---	------

A1	6	DAV
----	---	-----

A0	5	EOI
----	---	-----

D5	17	REN
----	----	-----

D7	10	SRQ
----	----	-----

D2	11	ATN
----	----	-----

A3	8	NDAC
----	---	------

D4	9	IFC
----	---	-----

```

D3    1    DIO1
D15   2    DIO2
D16   3    DIO3
D14   4    DIO4
D8     13   DIO5
D9     14   DIO6
D10    15   DIO7
D6     16   DIO8
GND    12   Shield
GND    18,19,20,21,22,23 GND

```

```
*/
```

```

#ifdef MICRO
#define APINS 4
#define DPINS 12
static const uint8_t analog_pins[] = {A0, A1, A2, A3};
static const uint8_t gpib_pins[] = {5, 6, 7, 8};
static const uint8_t digi[] = {1, 2, 3, 4, 9, 10, 11, 13, 14, 15, 16, 17};
static const uint8_t gpib_digi[] = {3, 15, 16, 14, 4, 7, 2, 8, 9, 10, 6, 5};
#endif

```

```

#ifdef UNO
#define APINS 5
#define DPINS 10
static const uint8_t analog_pins[] = {A0, A1, A2, A3, A4, A5};
static const uint8_t gpib_pins[] = {1, 2, 3, 4, 13, 14};
static const uint8_t digi[] = {5, 6, 7, 8, 9, 10, 11, 15, 16, 17};
static const uint8_t gpib_digi[] = {12, 11, 10, 9, 8, 2, 7, 4, 5, 3};
#endif

```

```

#define BAUDRATE 9200
#define PAUSE 1000 //delay after one test
#define TESTLOOP 20000
#define NOTIFY 5000 //in TESTLOOP if counter % NOTIFY == 0 then print "+"

```

```

void test(int odmin, int domax, uint8_t ap) {
    bool sm = true;
    int p;
    while (sm) {
        for (int i = 1; i <= TESTLOOP; i++) {
            p = analogRead(ap);
            if ((p >= odmin) && (p <= domax)) {
                sm = false;
            }
        }
    }
}

```

```

        if ((i % NOTIFY) == 0) {
            Serial.print("+");
        }
    } else {
        sm = true;
        break;
    }
}
}
}
Serial.println(" OK");
delay(PAUSE);
}

```

```

void testAnalogPin() {
    Serial.println("##### TEST analog pins");
#ifdef UNO
    pinMode(12, OUTPUT);
    digitalWrite(12, HIGH);
#endif
#ifdef MICRO
    pinMode(3, OUTPUT);
    digitalWrite(3, HIGH);
#endif
    for (int apin = 0; apin < APINS; apin++) {
        digitalRead(analog_pins[apin]);
        Serial.println("_____");
        Serial.print("GPIB pin >> ");
        Serial.print(gpib_pins[apin]);
        Serial.println(" <<");
        Serial.print(gpib_pins[apin]);
        Serial.print(" connect to GND ");
        test(0, 10, apin);
        Serial.print(gpib_pins[apin]);
#ifdef UNO
        Serial.print(" connect to pin 5 ");
#endif
#ifdef MICRO
        Serial.print(" connect to pin 1 ");
#endif
        test(1010, 1023, apin);
    }
}

```

```

void testDigitalPin() {
    Serial.println("##### TEST digital pins");
    for (int dp = 0; dp < DPINS; dp++) {
        pinMode(gpib_digi[dp], OUTPUT);
        Serial.println("\n*****");
    }
}

```

```

    Serial.print("digital pin >> ");
    Serial.print(digi[dp]);
    Serial.println(" << HIGH test");
    digitalWrite(gpib_digi[dp], HIGH);
    Serial.print(digi[dp]);

#ifdef UNO
    Serial.print(" connect to pin 1 ");
#endif
#ifdef MICRO
    Serial.print(" connect to pin 5 ");
#endif

    test(1013, 1023, A0);
    Serial.print(digi[dp]);
#ifdef UNO
    Serial.print(" connect to pin 2 ");
#endif
#ifdef MICRO
    Serial.print(" connect to pin 6 ");
#endif

    test(1013, 1023, A1);
    Serial.print("\ndigital pin >> ");
    Serial.print(digi[dp]);
    Serial.println(" << LOW test");
    digitalWrite(gpib_digi[dp], LOW);
    Serial.print(digi[dp]);
#ifdef UNO
    Serial.print(" connect to pin 1 ");
#endif
#ifdef MICRO
    Serial.print(" connect to pin 5 ");
#endif

    test(0, 10, A0);
    Serial.print(digi[dp]);
#ifdef UNO
    Serial.print(" connect to pin 2 ");
#endif
#ifdef MICRO
    Serial.print(" connect to pin 6 ");
#endif

    test(0, 10, A1);
}
}

```

```

void setup()
{
  Serial.begin(BAUDRATE);
  while (!Serial);
  for (int i = 1; i <= 30; i++) {
    Serial.println("\n\n\n");
  };
  Serial.println("#####");
  Serial.println("####");
  Serial.println("####   GPIB AR488 wiring test");
  Serial.println("####");
#ifdef UNO
  Serial.println("####   Arduino UNO / Nano ORIGINAL only!");
#endif
#ifdef MICRO
  Serial.println("####");
  Serial.println("####   Arduino Pro Micro");
#endif
  Serial.println("####");
  Serial.println("#####\n\n");
}

void loop()
{
  testAnalogPin();
  testDigitalPin();
  Serial.println("\n\n---- TEST DONE ----");

  while (true) { //endless loop
  }
}

```