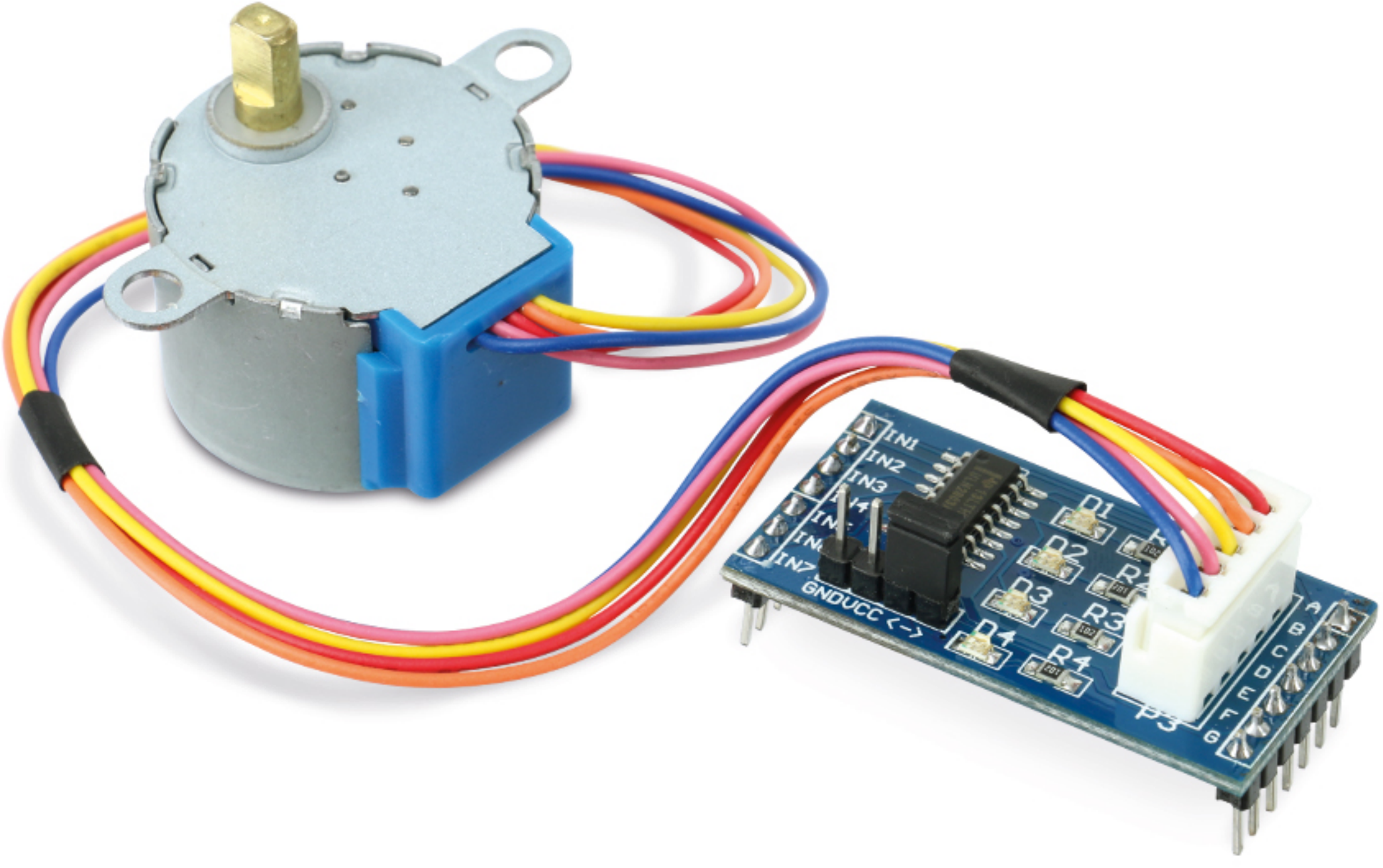
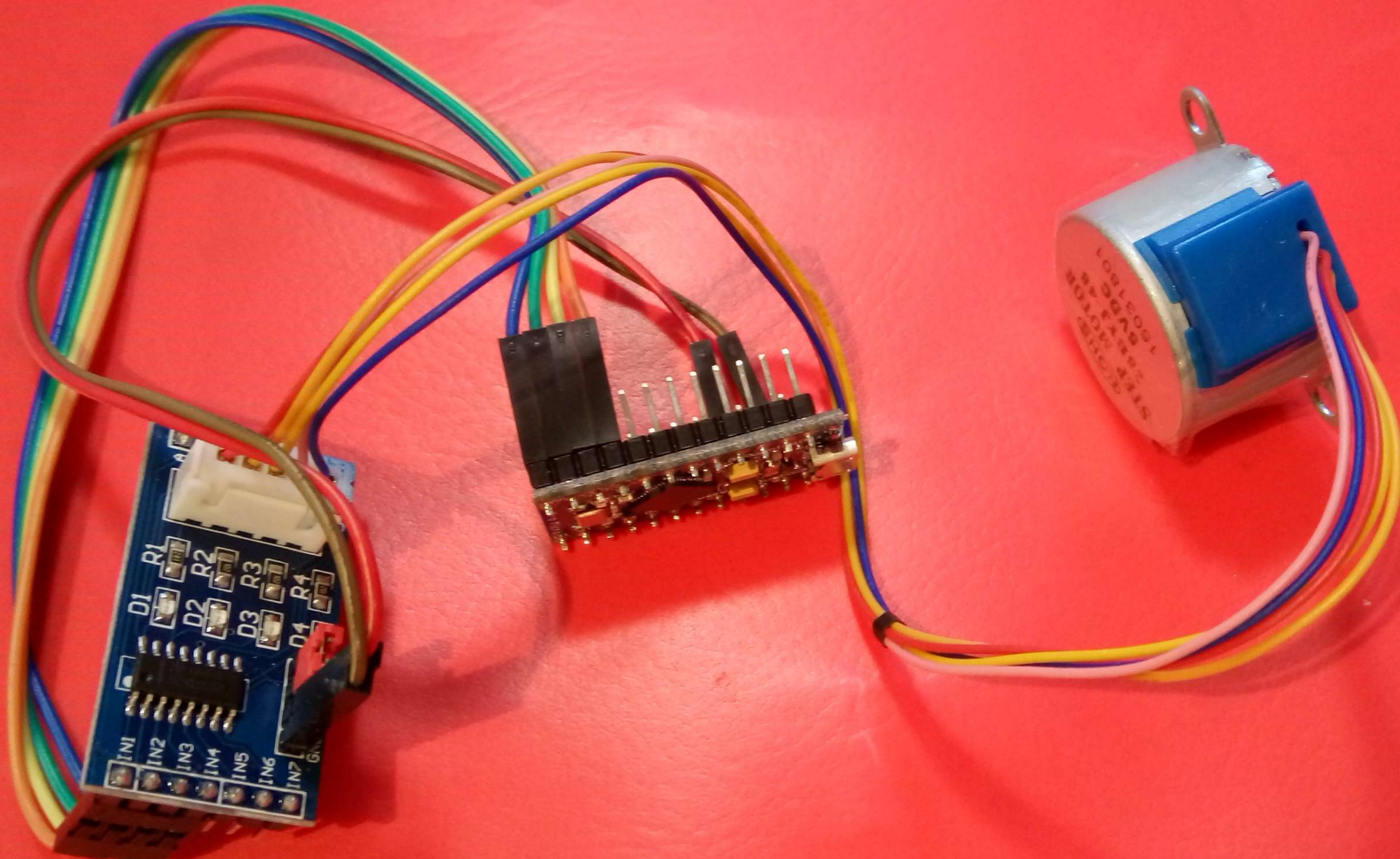
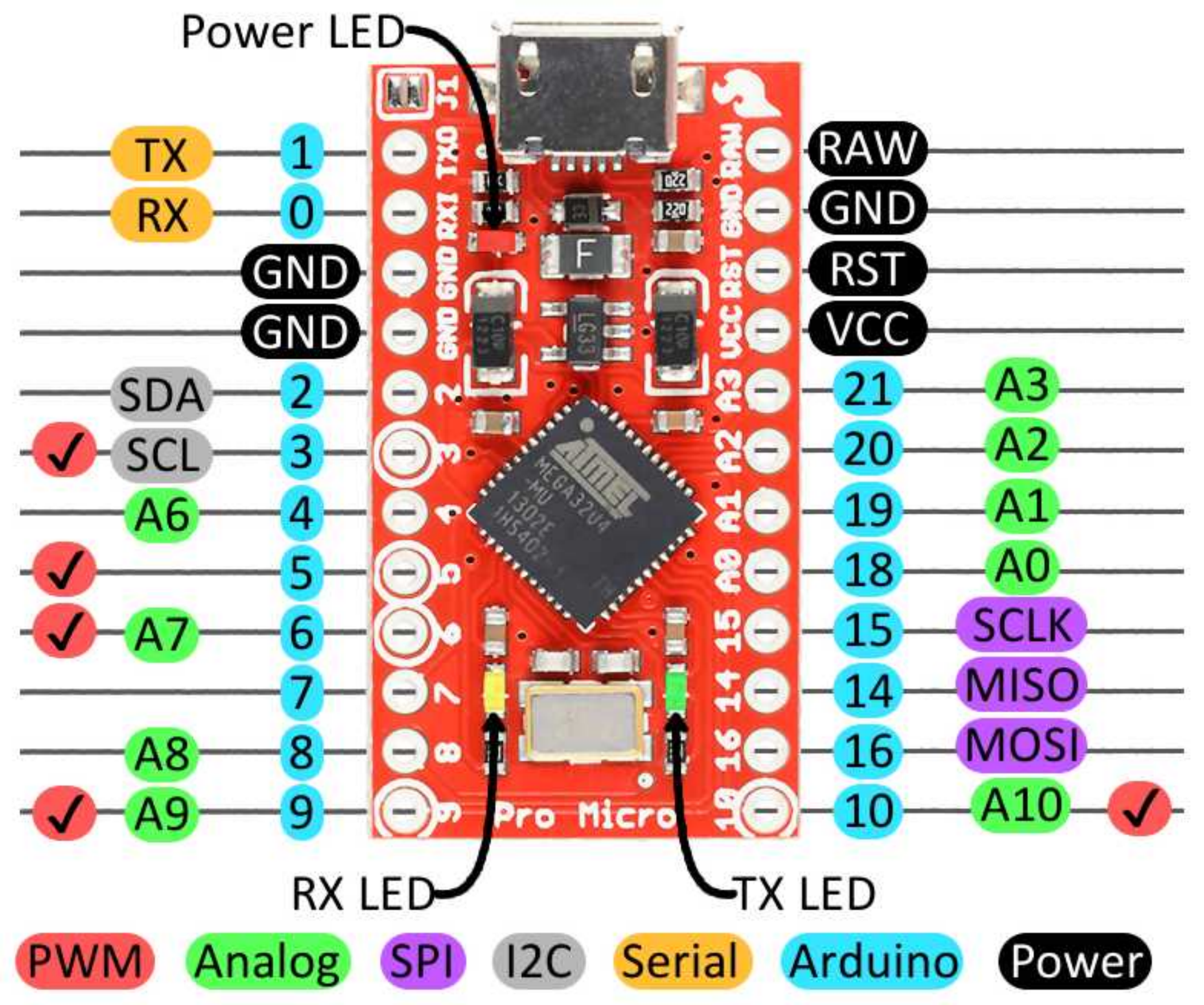

IN1
IN2
IN3
IN7
GNDUCC
D1
D2
D3
D4
R1
R3
P3
A
B
C
D
E
G


R1
R2
R3
R4
D1
D2
D3
D4
IN1
IN2
IN3
IN4
IN5
IN6
IN7


Power LED
TX
RX
GND
GND
SDA
SCL
A6
A7
A8
A9
1
0
2
3
4
5
6
7
8
9
RAW
GND
RST
VCC
21
20
19
18
15
14
16
10
A3
A2
A1
A0
SCLK
MISO
MOSI
A10
RX LED
TX LED
Pro Micro
PWM
Analog
SPI
I2C
Serial
Arduino
Power

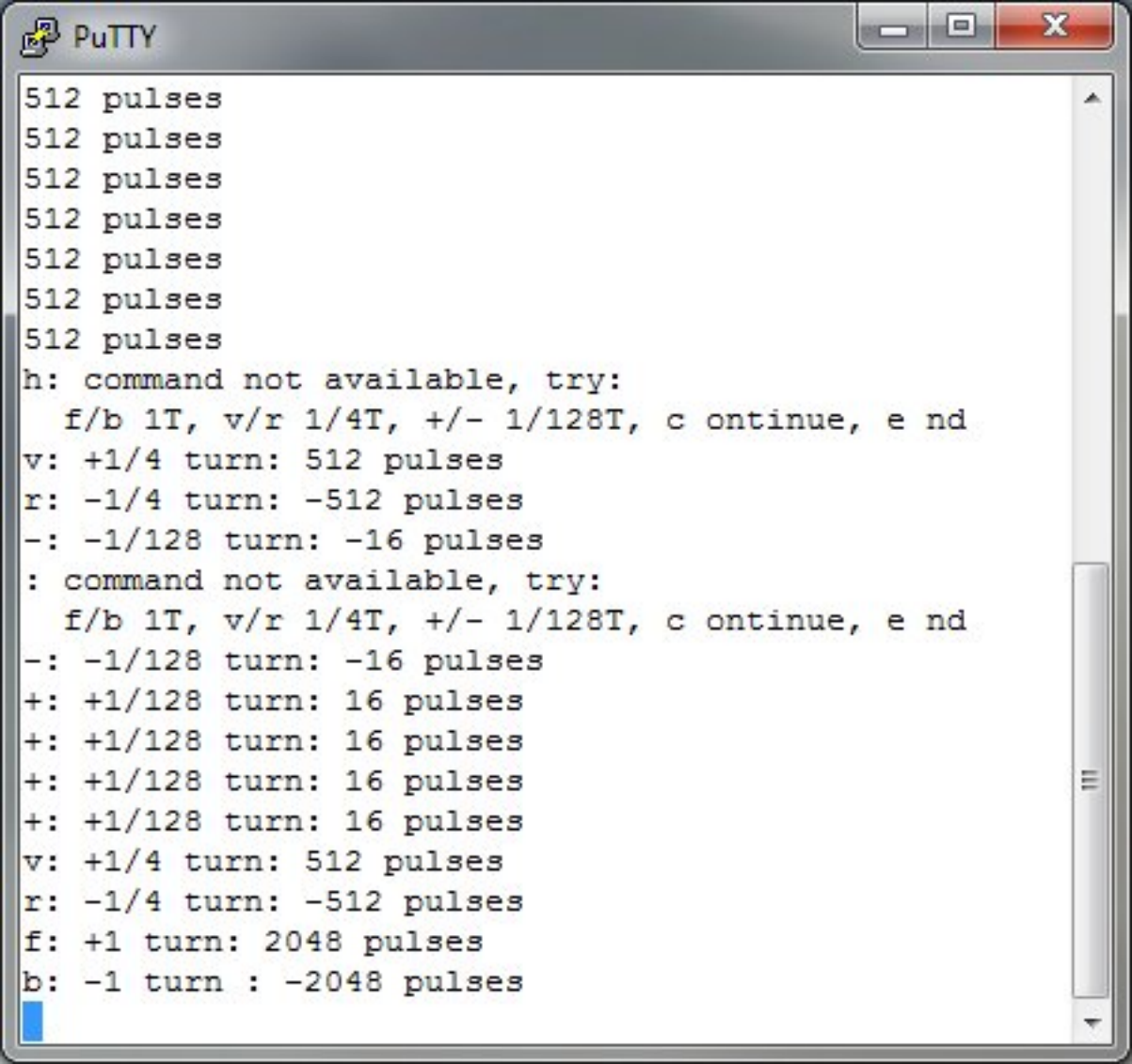
```
/*
 * UART controlled stepper servo
 * for Arduino pro micro ATmega32U4, UART 115200 Baud, USB-VCP 8n1n
 * Pollin Schrittmotor-Set S-SPSM-5V, Best.Nr. 310 543, price 4,16 €
 * Motor 28BYJ-48 mit 4 Windungen: 1-5-3, 2-5-4. 5:VDD
 * Achtung: der Motor braucht 2048 Pulse pro Umdrehung
  *http://www.arduino.cc/en/Reference/Stepper
 * This example code is in the public domain.
 * G. Heinz
 * Motor will be initialized with 'c' (continous run)
 * set "Tools/Serial Monitor/115200 Baud",
 * type 'h' for help
 * proMicro internal LEDs at 8: PB0 and 22: PD5
 */
#include <Stepper.h>

// set the number of steps of the motor
#define STEPS 2048

// define the pins of the motor
// Motor 28BYJ-48 Windungen: Pins 1-5-3, 2-5-4. 5:VDD
// Evtl. mit Logikanalysator Phase prüfen, Ziel: pin1 = /pin3, pin2 = /pin4
#define pin1 6
#define pin2 8
#define pin3 7
#define pin4 9
```

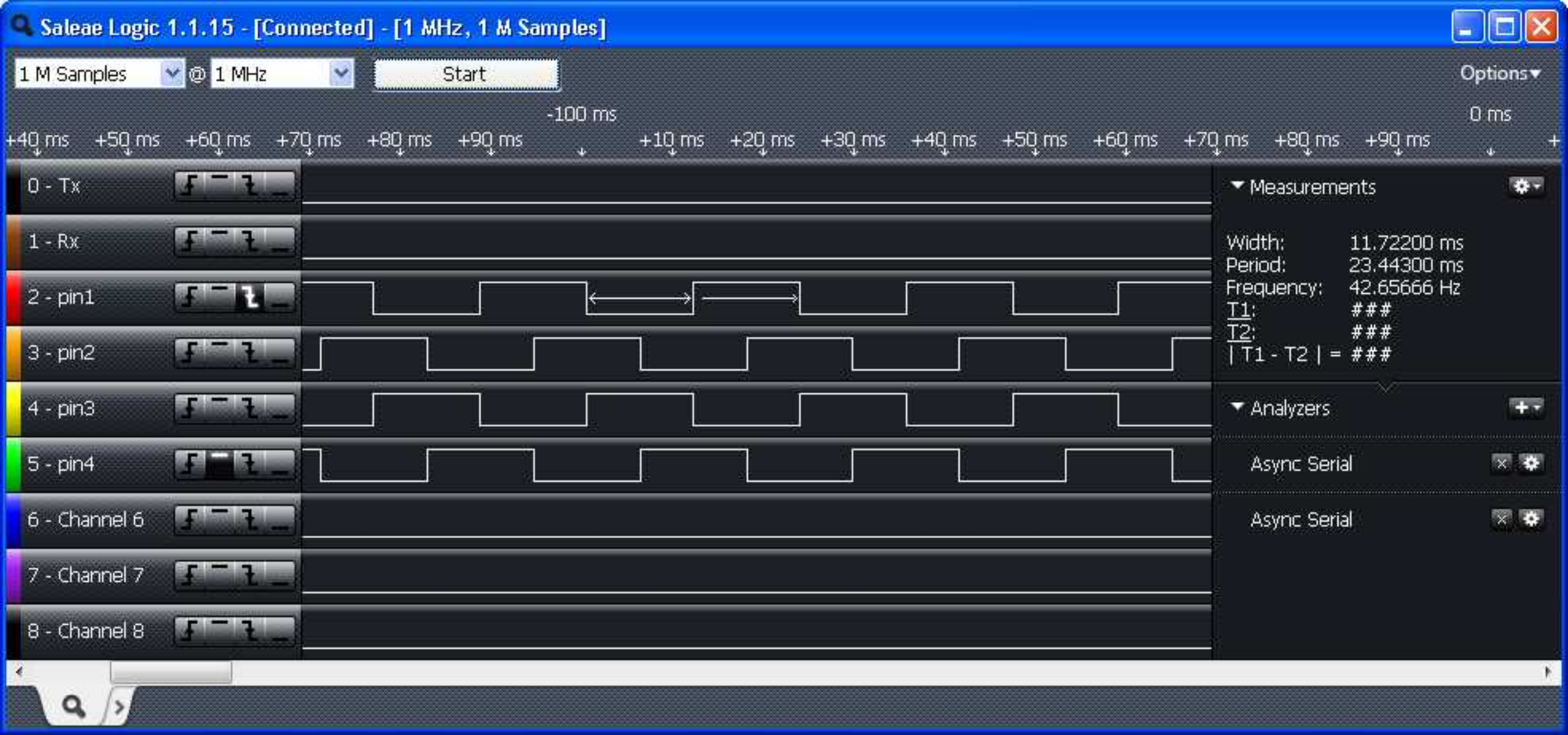
```
int val;
char  ch  ='c';

// create an instance of the stepper class
Stepper stepmotor(STEPS, pin1, pin2, pin3, pin4);// wndg groups 6-8 and 7-9 (!)

// bei Stillstand Strom durch alle Windungen abschalten
void standby(void)   {// Motor soll nicht heiß werden
  digitalWrite(pin1,0);
  digitalWrite(pin2,0);
  digitalWrite(pin3,0);
  digitalWrite(pin4,0);
}


////////////////////////////////////////////////////////////////

void setup() {
  // USB-VCP eröffnen
  Serial.begin(115200);

  // set the speed of the motor in rpm (Runden pro Minute)
  // maximal 5 rpm, sonst Anlaufprobleme
  stepmotor.setSpeed(5);    // führt auf 42,6Hz
}

void loop() {
```

```
// read UART
while (Serial.available() > 0) {
   ch =Serial.read();
  }

// decode receiving UART character
switch   (ch)   {
  case 'f': val =  2048; ch='\0'; Serial.print("f: +1 turn: ");       break;   //
  case 'b': val = -2048; ch='\0'; Serial.print("b: -1 turn : ");      break;   //
  case 'v': val =  512;  ch='\0'; Serial.print("v: +1/4 turn: ");     break;   //
  case 'r': val = -512;  ch='\0'; Serial.print("r: -1/4 turn: ");     break;   //
  case '+': val =  16;   ch='\0'; Serial.print("+: +1/128 turn: ");   break;   //
  case '-': val = -16;   ch='\0'; Serial.print("-: -1/128 turn: ");   break;   //
  case 'c':          val          =          512;break;   // run continous forward
  case 'e':  val = 0;    ch='\0'; Serial.println("end of rotation "); break;   //
  case '\0': val = 0;    ch='\0';               break;   // reset, nichts tun
  default:              {// falscher Befehl. Hilfe zeigen:
    Serial.print(ch); Serial.println(": command not available, try: ");
    Serial.println("  f/b 1T, v/r 1/4T, +/- 1/128T, c ontinue, e nd");
     val = 0; ch ='\0';   // reset
   }break;
}

// protocol printout
if (val != 0)  {
  Serial.print(val);
  Serial.println(" pulses");
```

```
  }

  // move the number of steps
  stepmotor.step(val);

  // go into standby state
  if (ch !='c') {
    delay(10);          // warte auf Stillstand
    standby();
  }
}

/*  UART DIALOG
.....
512 pulses
512 pulses
512 pulses
h: command not available, try:
  f/b 1T, v/r 1/4T, +/- 1/128T, c ontinue, e nd
-: -1/128 turn: -16 pulses
-: -1/128 turn: -16 pulses
-: -1/128 turn: -16 pulses
-: -1/128 turn: -16 pulses
-: -1/128 turn: -16 pulses
-: -1/128 turn: -16 pulses
v: +1/4 turn: 512 pulses
512 pulses
```

```
512 pulses
512 pulses
512 pulses
end of rotation
+: +1/128 turn: 16 pulses
-: -1/128 turn: -16 pulses
+: +1/128 turn: 16 pulses
v: +1/4 turn: 512 pulses
v: +1/4 turn: 512 pulses
h: command not available, try:
  f/b 1T, v/r 1/4T, +/- 1/128T, c ontinue, e nd
+: +1/128 turn: 16 pulses
v: +1/4 turn: 512 pulses
v: +1/4 turn: 512 pulses
v: +1/4 turn: 512 pulses
-: -1/128 turn: -16 pulses
-: -1/128 turn: -16 pulses
-: -1/128 turn: -16 pulses
512 pulses
512 pulses
512 pulses
end of rotation
.....
*/
```


PuTTY
512 pulses
512 pulses
512 pulses
512 pulses
512 pulses
512 pulses
512 pulses
h: command not available, try:
f/b 1T, v/r 1/4T, +/- 1/128T, c ontinue, e nd
v: +1/4 turn: 512 pulses
r: -1/4 turn: -512 pulses
-: -1/128 turn: -16 pulses
: command not available, try:
f/b 1T, v/r 1/4T, +/- 1/128T, c ontinue, e nd
-: -1/128 turn: -16 pulses
+: +1/128 turn: 16 pulses
+: +1/128 turn: 16 pulses
+: +1/128 turn: 16 pulses
+: +1/128 turn: 16 pulses
v: +1/4 turn: 512 pulses
r: -1/4 turn: -512 pulses
f: +1 turn: 2048 pulses
b: -1 turn : -2048 pulses


Saleae Logic 1.1.15 - [Connected] - [1 MHz, 1 M Samples]
1 M Samples
@ 1 MHz
Start
Options
-100 ms
0 ms
+40 ms
+50 ms
+60 ms
+70 ms
+80 ms
+90 ms
+10 ms
+20 ms
+30 ms
+40 ms
+50 ms
+60 ms
+70 ms
+80 ms
+90 ms
0 - Tx
1 - Rx
2 - pin1
3 - pin2
4 - pin3
5 - pin4
6 - Channel 6
7 - Channel 7
8 - Channel 8
Measurements
Width: 11.72200 ms
Period: 23.44300 ms
Frequency: 42.65666 Hz
T1: ###
T2: ###
| T1 - T2 | = ###
Analyzers
Async Serial
Async Serial