

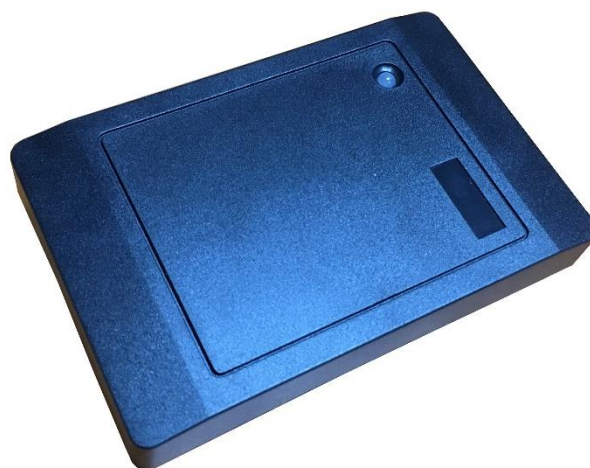
RFID čtečka 125 kHz

1. POPIS

Čtečka je určena pro čtení 125 kHz RFID čipů. Systém je uložen v plastovém boxu a PCB je chráněna epoxidovou pryskyřicí. Elektronika je doplněna o indikační LED diodu a bzučák. Zařízení čte 26 bitové ID čipů v binární formě.

Základní charakteristika:

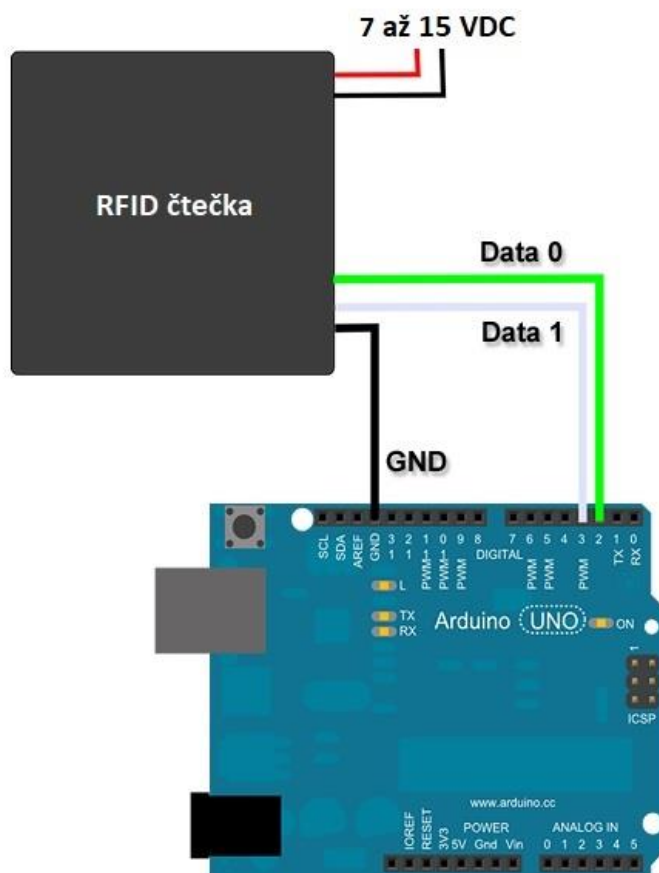
- napájení 7 až 15 V DC
- kompatibilní s platformou Arduino
- komunikační rozhraní Wiegand 26 bitů
- voděodolné pouzdro



2. SPECIFIKACE

Komunikační rozhraní	Wiegand 26 nebo 34 bitů	Dosah čtení	až 15 cm
Frekvence čipů	125 kHz	Doba rozpoznání čipu	<200 ms
Podpor. typ čipů	EM	Délka čtecí interval	<500 ms
Napájení	7 až 15 V DC	Rozteč mont. otvorů (mm)	85
Pracovní proud	až 70 mA	Rozměry mont. otvorů (mm)	7 x 3,5
Max. délka vedení	60 až 100 m	Rozměry (mm)	115 x 74 x 16

3. ZAPOJENÍ



Popis vodičů

Barva vodiče	Popis
Červená	Napájecí svorka +, 7 až 15 V DC
Černá	Napájecí svorka –, GND
Zelená	Datový vodič Wiegand D0
Bílá	Data vodič Wiegand D1
Šedá	Aktivace bzučáku, aktivní při propojení s GND (log. 0)
Fialová	Přepínání formátu výstupních dat. Při propojení s GND je výstupní formát Wiegand 34bit, při nezapojení je formát Wiegand 26bit.

00101
01001
00001

4. UKÁZKA PROGRAMU

Kód je převzat z příkladů knihovny [WiegandNG.h](#). Přechtené ID čipu je zobrazeno na sériové lince v binární formě.

```
#include <WiegandNG.h>

WiegandNG wg;

void PrintBinary(WiegandNG &tempwg) {
    volatile unsigned char *buffer=tempwg.getRawData();
    unsigned int bufferSize = tempwg.getBufferSize();
    unsigned int countedBits = tempwg.getBitCounted();

    unsigned int countedBytes = (countedBits/8);
    if ((countedBits % 8)>0) countedBytes++;
    // unsigned int bitsUsed = countedBytes * 8;

    for (unsigned int i=bufferSize-countedBytes; i < bufferSize;i++) {
        unsigned char bufByte=buffer[i];
        for(int x=0; x<8;x++) {
            if ( (((bufferSize-i) *8)-x) <= countedBits) {
                if((bufByte & 0x80)) {
                    Serial.print("1");
                }
                else {
                    Serial.print("0");
                }
            }
            bufByte<<=1;
        }
        Serial.println();
    }
}

void setup() {
    Serial.begin(9600);

    // for UNO just use wg.begin(), will default to Pin 2 and Pin 3 connected to D0 and D1 respectively
    // initialize Wiegand ND for 48 bit data, every 8 bits take up 1 byte of Arduino memory
    // as long as there is still memory, user can even capture 1024 bit Wiegand by calling wg.begin(1024)

    unsigned int pinD0 = 2;
    unsigned int pinD1 = 3;
    unsigned int wiegandbits = 48;
    unsigned int packetGap = 15; // 25 ms between packet

    if(!wg.begin(pinD0, pinD1, wiegandbits, packetGap)) {
        Serial.println("Out of memory!");
    }
    Serial.println("Ready...");
}

void loop() {
    if(wg.available()) {
```

```
interrupts      wg.pause(); // pause Wiegand pin
                Serial.print("Bits=");
                Serial.println(wg.getBitCounted()); // display the number of bits counted
                Serial.print("RAW Binary=");
                PrintBinary(wg); // display raw data in binary form,
raw data inclusive of PARITY
                wg.clear(); // compulsory to call
clear() to enable interrupts for subsequent data
                }
}
```