

## RFID čtečka 13,56 MHz

### 1. POPIS

Čtečka je určena pro čtení RFID čipů typu Mifare (13,56 MHz). Systém je uložen v plastovém boxu a PCB je chráněna epoxidovou pryskyřicí. Elektronika je doplněna o indikační LED diodu a buzzer. Zařízení čte 26 bitové ID čipů v binární formě.

Základní charakteristika:

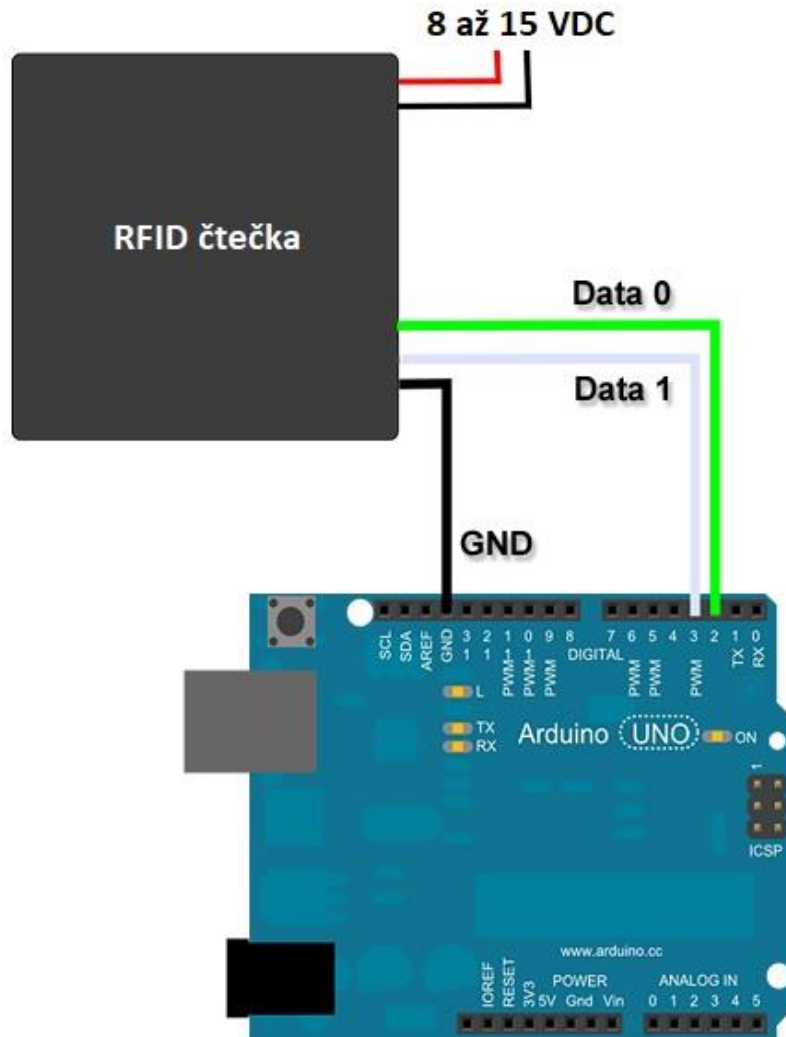
- napájení 8 až 15 V
- kompatibilní s platformou Arduino
- komunikační rozhraní Wiegand 26 bitů
- voděodolné pouzdro



### 2. SPECIFIKACE

<b>Komunikační rozhraní</b>	Wiegand 26 bitů	<b>Max. délka vedení</b>	60 až 100 m
<b>Frekvence</b>	13,56 MHz	<b>Dosah čtení</b>	až 15 cm
<b>Pro typ čipů</b>	Mifare	<b>Rozpoznání čipu</b>	<200 ms
<b>Napájení</b>	8 až 15 V	<b>Čtecí interval</b>	<500 ms
<b>Pracovní proud</b>	až 70 mA	<b>Rozměry (mm)</b>	115 x 74 x 16

### 3. ZAPOJENÍ



### 00101 01001 00001 4. UKÁZKA PROGRAMU

Kód je převzat z příkladů knihovny [WiegandNG.h](http://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()) {
        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
    }
}

```