

eses výukový multi shield pro Arduino UNO



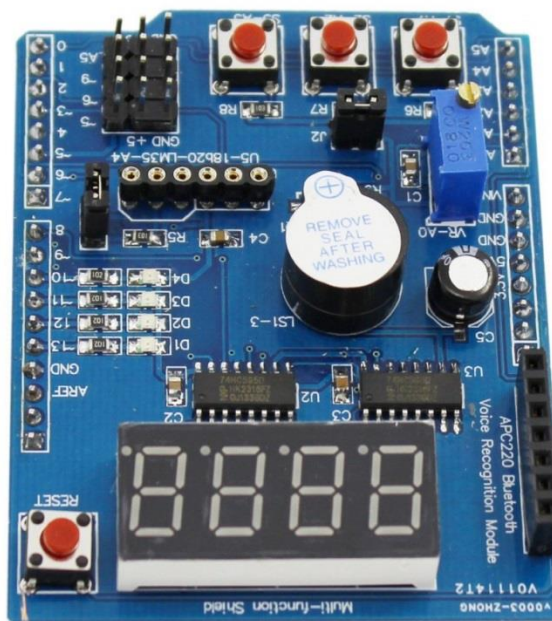
1. POPIS

Tento shield je vhodný především pro výuku a začátečníky, jelikož obsahuje spoustu základních periférií, které se nadále využívají v pokročilejších projektech.

Díky shieldu Funduino je možno snadno demonstrovat použití AD převodníku, zobrazování hodnot na displeji, řízení pohybu servomotoru a spoustu dalších funkcí. Shield je kompatibilní s vývojovými kity Arduino UNO, Leonardo, Duemilanove, Diecimila, ADK/Mega R3 a vyššími, přes které je i napájen.

Shield obsahuje:

- 4x indikační LED
- 3x tlačítko + tlačítko reset
- Bzučák
- 7segmentový displej s čipem 74HC595
- Víceotáčkový odporový trimr (pin A0)
- Patice pro teploměr LM35, teploměr DS18B20, pro BT a jiné komunikační moduly, pro servo





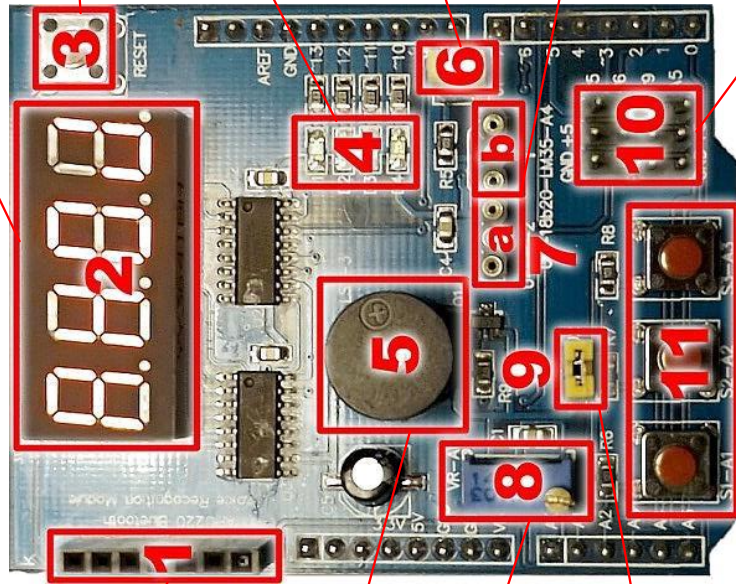
2. ZAPOJENÍ A PINOUT

K zapojení nejsou zapotřebí žádné externí komponenty, shield připojíme přímo k vývojovému kitu.

4 červené LED diody	10, 11, 12, 13	Konektor pro IR senzor	2
3 tlačítka	A1, A2, A3	Konektor pro teploměr	A4
Trimr (10 kOhm)	A0	Konektor pro dig. senzory	0 (Rx), 1 (Tx)
LED displej řízený 74HC595	4, 7, 8	Konektor pro servomotory (řídící piny)	5, 6, 9, A5
Buzzer/sirénka	3		

Arduino Multi-function Shield

(Funduino výukový multi-shield do škol pro Arduino)



Konektor pro modul Bluetooth APC220
(nebo Voice Recognition Module)
Popis (směrem shora)

- 7 – nepoužito
- 6 – nepoužito
- 5 – pin 0
- 4 – pin 1
- 3 – nepoužito
- 2 – napětí +5 V
- 1 – GND

Piezo-bzučák pasivní – připojen k pinu 3 (PWM)

Odporový trimr 10 kΩ mezi referenční napětí (+5 V) a GND, jezdec vyveden na pin A0

Propojka (jumper) – připojuje Pull Up rezistory (10 kΩ) k tlačítkům S1, S2 a S3 (viz 11)

Vstupní tlačítka – připojena k pinům A1-3 (stisknutím – hodnota LOW)

- Popis (zleva)*
- Tlačítko S1 – pin A1
 - Tlačítko S2 – pin A2
 - Tlačítko S3 – pin A3

Čtevice LED segmentů řízených obvodem 74HC595
Zápis do Latch – pin 4 (start zápisu LOW, konec HIGH)
Hodinový pulz (takt) – pin 7

Sériová data – pin 8 – zápis 2x 8 bitů (2x Byte)

(první Byte – data rozsvícených LED – rozsvícení LOW, vypnutí HIGH)



Bity MSB LSB odpovídají segmentu:



MSB LSB odpovídají segmentu:

V druhém Byte se zapisuje adresa:

První (zleva) zobrazovač: 0000001

Druhý (zleva) zobrazovač: 0000010

Třetí (zleva) zobrazovač: 0000100

Čtvrtý (zleva) zobrazovač: 0001000

Reset – resetuje modul Arduino (nikoliv Shield)

Signalizační LED (rozsvícení LOW, vypnutí HIGH)

Popis LED (směrem shora)

LED D1 – pin 13 (PWM)

LED D2 – pin 12

LED D3 – pin 11 (PWM)

LED D4 – pin 10 (PWM)

Propojka (jumper) – připojuje Pull Up rezistor 10 kΩ na vstup A4 (vyvedeno na střední pin v konektoru 7b)

Konektor pro připojení čidla IR (7a) a teplotných čidel Dallas 18B20 nebo LM35 (7b)

Popis 7a (zleva)

1 – pin 2

2 – GND

3 – napětí +5 V

Popis 7b (zleva)

1 – GND

2 – pin A4

3 – napětí +5 V

Konektor pro připojení čidel (např. sonar HCSR04, servo...)

Popis (zleva)

První sloupec – GND (všechny 4 piny)

Druhý sloupec – napětí +5 V (všechny 4 piny)

Třetí sloupec (shora)

pin 5 (PWM),

pin 6 (PWM),

pin 9 (PWM)

pin A5

Vytvořil: RNDr. Miroslav Panoš, Ph.D.



4. UKÁZKA PROGRAMU

Ukázka je převzata z navody.arduino-shop.cz/navody-k-produktum/funduino-vyukovy-multi-shield.html

```
// Arduino Multifunkční shield

// nastavení propojovacích pinů pro segmentový displej

#define pinLatch 4

#define pinClk 7

#define pinData 8

// nastavení propojovacích pinů pro trimr

#define trimr A0

// nastavení propojovacích pinů pro LED diody

const byte LED[] = {13,12,11,10};

// nastavení propojovacích pinů pro tlačítka

#define tlacitkoA1 A1

#define tlacitkoA2 A2

#define tlacitkoA3 A3

// nastavení propojovacích pinů pro sirénku

#define pinSirenka 3

// definování stavů pro vypnutí a zapnutí

#define ON LOW

#define OFF HIGH

// proměnná pro počítadlo stisku tlačítek

int pocitadlo = 0;

// mapa čísel pro segmentový displej - čísla 0 až 9

const byte mapaSegment[] =
{0xC0,0xF9,0xA4,0xB0,0x99,0x92,0x82,0xF8,0X80,0X90};

// mapa číslic na segmentovém displeji
```

```

const byte mapaCisloSeg[] = {0xF1,0xF2,0xF4,0xF8};

void setup () {
    // nastavení propojovacích pinů pro
    // segmentový displej, LED diody
    // a sirénku jako výstup
    pinMode(pinLatch, OUTPUT);
    pinMode(pinClk, OUTPUT);
    pinMode(pinData, OUTPUT);
    pinMode(LED[0], OUTPUT);
    pinMode(LED[1], OUTPUT);
    pinMode(LED[2], OUTPUT);
    pinMode(LED[3], OUTPUT);
    pinMode(pinSirenka, OUTPUT);
    // vypnutí všech LED diod a sirénky
    digitalWrite(LED[0], OFF);
    digitalWrite(LED[1], OFF);
    digitalWrite(LED[2], OFF);
    digitalWrite(LED[3], OFF);
    noTone(pinSirenka);
}

void loop() {
    /* práce se segmentovým displejem a trimrem */
    // vytvoření proměnné pro načtení analogového vstupu z trimru
    int analogTrimr = analogRead(trimr);
    // zapsání analogové hodnoty na segmentový displej,
    // zápis je po číslech - tisícovky, stovky, ..
    zapisCisloNaSegment(0 , analogTrimr / 1000);
}

```

```

zapisCisloNaSegment(1 , (analogTrimr / 100) % 10);
zapisCisloNaSegment(2 , (analogTrimr / 10) % 10);
zapisCisloNaSegment(3 , analogTrimr % 10);
/* obsluha tlačítek, LED diod a sirénky */
// načtení stavu tlačítek do proměnných
bool tlac1 = digitalRead(tlacitkoA1);
bool tlac2 = digitalRead(tlacitkoA2);
bool tlac3 = digitalRead(tlacitkoA3);
// pokud je stisknuto první tlačítko, přičteme jedničku
// do počítadla a zobrazíme na LED diodách v binárním tvaru
if (tlac1 == ON) {
    pocitadlo++;
    // na 4 LED diody lze vypsat maximálně číslo 15 v bin. tvaru
    if (pocitadlo > 15) {
        pocitadlo = 15;
    }
    // zápis pomocí funkce bitRead(proměnná, pozice bitu)
    // nutné znegovat pomocí ! - opačná logika zapínání LED diod
    digitalWrite(LED[0], !(bitRead(pocitadlo, 0)));
    digitalWrite(LED[1], !(bitRead(pocitadlo, 1)));
    digitalWrite(LED[2], !(bitRead(pocitadlo, 2)));
    digitalWrite(LED[3], !(bitRead(pocitadlo, 3)));
    // pauza pro plynulejší přičítání mezi stisky tlačítek
    delay(200);
}
// pokud je stisknuto druhé tlačítko, odečteme jedničku
// z počítadla, zbytek je stejný jako u tlačítka 1
if (tlac2 == ON) {

```

```

pocitadlo--;

if (pocitadlo < 0) {
    pocitadlo = 0;
}

digitalWrite(LED[0], !(bitRead(pocitadlo, 0)));
digitalWrite(LED[1], !(bitRead(pocitadlo, 1)));
digitalWrite(LED[2], !(bitRead(pocitadlo, 2)));
digitalWrite(LED[3], !(bitRead(pocitadlo, 3)));

delay(200);
}

// pokud je stisknuto třetí tlačítko, zahrajeme tón
if (tlac3 == ON) {
    // hraný tón začíná od komorního A a se zvyšujícím
    // se počítadlem zvyšujeme frekvenci tónu po 100 Hz
    int ton = 440 + (pocitadlo * 100);

    // zahrání tónu na sirénce po dobu 20 milisekund
    tone(pinSirenka, ton, 20);
}
}

// podprogram pro zápis čísla na zadaný segment
void zapisCisloNaSegment(byte segment, byte hodnota) {
    digitalWrite(pinLatch, LOW);

    shiftOut(pinData, pinClk, MSBFIRST, mapaSegment[hodnota]);
    shiftOut(pinData, pinClk, MSBFIRST, mapaCisloSeg[segment] );

    digitalWrite(pinLatch, HIGH);
}

```