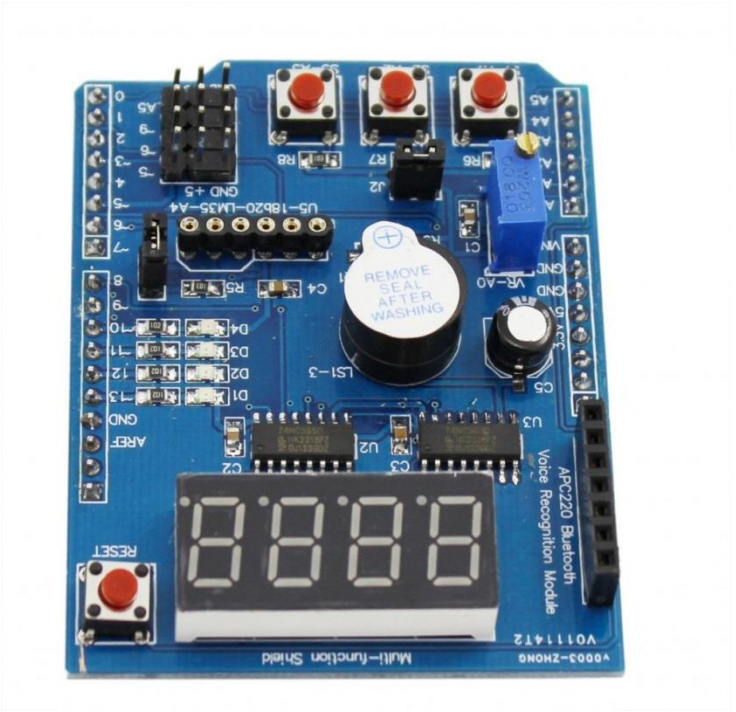
Funduino výukový multi shield do škol pro Arduino

1. POPIS

Tento shield je vhodný především pro výuku a začátečníky, jelikož obsahuje spoustu základních periferií, 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

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

3. UKÁZKA PROGRAMU

Ukázky jsou převzaty z http://arduinolearning.com/code/multi-function-shield-examples.php

Blikání LED

```
int led1 = 13;
int led2 = 12;
int led3 = 11;
int led4 = 10;

void setup()
{
```

ECLIPSERA s.r.o. Distributor pro ČR.

Copyright © 2016 ECLIPSERA s.r.o.

```
// initialize the digital pin as an output.
pinMode(led1, OUTPUT);
pinMode(led2, OUTPUT);
pinMode(led3, OUTPUT);
pinMode(led4, OUTPUT);
}

void loop()
{
digitalWrite(led1, HIGH);
digitalWrite(led2, HIGH);
digitalWrite(led3, HIGH);
digitalWrite(led4, HIGH);
delay(1000);
digitalWrite(led1, LOW);
digitalWrite(led2, LOW);
digitalWrite(led3, LOW);
digitalWrite(led4, LOW);
delay(1000);
}
```

Tlačítka

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

#define BUTTON1 A1
#define BUTTON2 A2

void setup()
{
  // initialize the digital pin as an output.
  /* Set each pin to outputs */
  pinMode(LED[0], OUTPUT);
  pinMode(LED[1], OUTPUT);
  pinMode(LED[2], OUTPUT);
  pinMode(LED[3], OUTPUT);
}

void loop()
{
  if (!digitalRead(BUTTON1))
  {
    digitalWrite(LED[0], HIGH);
    digitalWrite(LED[1], HIGH);
    digitalWrite(LED[2], HIGH);
    digitalWrite(LED[3], HIGH);
  }

  if (!digitalRead(BUTTON2))
  {
    digitalWrite(LED[0], LOW);
    digitalWrite(LED[1], LOW);
    digitalWrite(LED[2], LOW);
    digitalWrite(LED[3], LOW);
  }
}
```

Potenciometr

```
#define Pot1 0

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

/* Main Program */
void loop()
{
```


Copyright © 2016 ECLIPSERA s.r.o.

```
  Serial.print("Potentiometer reading: ");
  Serial.println(analogRead(Pot1));
  /* Wait 0.5 seconds before reading again */
  delay(500);
}
```

Displej

```
/* Define shift register pins used for seven segment display */
#define LATCH_DIO 4
#define CLK_DIO 7
#define DATA_DIO 8

/* Segment byte maps for numbers 0 to 9 */
const byte SEGMENT_MAP[] = {0xC0, 0xF9, 0xA4, 0xB0, 0x99, 0x92, 0x82, 0xF8, 0X80, 0X90};
/* Byte maps to select digit 1 to 4 */
const byte SEGMENT_SELECT[] = {0xF1, 0xF2, 0xF4, 0xF8};

void setup ()
{
  /* Set DIO pins to outputs */
  pinMode(LATCH_DIO, OUTPUT);
  pinMode(CLK_DIO, OUTPUT);
  pinMode(DATA_DIO, OUTPUT);
}

/* Main program */
void loop()
{

  /* Update the display with the current counter value */
  WriteNumberToSegment(0 , 0);
  WriteNumberToSegment(1 , 1);
  WriteNumberToSegment(2 , 2);
  WriteNumberToSegment(3 , 3);
}

/* Write a decimal number between 0 and 9 to one of the 4 digits of the display */
void WriteNumberToSegment(byte Segment, byte Value)
{
  digitalWrite(LATCH_DIO, LOW);
  shiftOut(DATA_DIO, CLK_DIO, MSBFIRST, SEGMENT_MAP[Value]);
  shiftOut(DATA_DIO, CLK_DIO, MSBFIRST, SEGMENT_SELECT[Segment] );
  digitalWrite(LATCH_DIO, HIGH);
}
```


Copyright © 2016 ECLIPSERA s.r.o.