

Gymnázium a Střední průmyslová škola elektrotechniky a informatiky,
Frenštát pod Radhoštěm, příspěvková organizace

Projekt předmětu Praxe

Bezdrátový měřič vzdálenosti

Alois Miltak

E3C

28. října 2024

Zadání

Vytvořte program a zrealizujte zapojení pro Arduino, které pomocí ultrazvukového senzoru měří vzdálenost objektu a posílá tuto hodnotu v centimetrech na sériový port. Dále rozsvěcí LEDky a mění zvuk pieza v závislosti na změřené vzdálenosti.

Popis řešení

Nadefinoval jsem si čísla všech pinů jako konstanty, čísla pinů LEDek jsem si uložil do pole, abych mohl snadno LEDky nastavovat cyklem. Pro měření vzdálenosti jsem použil algoritmus z tutorialu uvedeného ve zdrojích. Limity vzdálenost, od kterých se rozsvěcují jednotlivé LEDky jsem si uložil do pole, abych nemusel psát pod sebou několik if-else, ale mohl vše nastavit jedním cyklem.

Použité součástky

- Arduino UNO
- Ultrazvukový senzor HC-SR04
- Piezo
- LEDky (2xčervená, 2x žlutá, 2x zelená)
- Rezistory 330Ω
- Nepájivé pole

Schéma zapojení

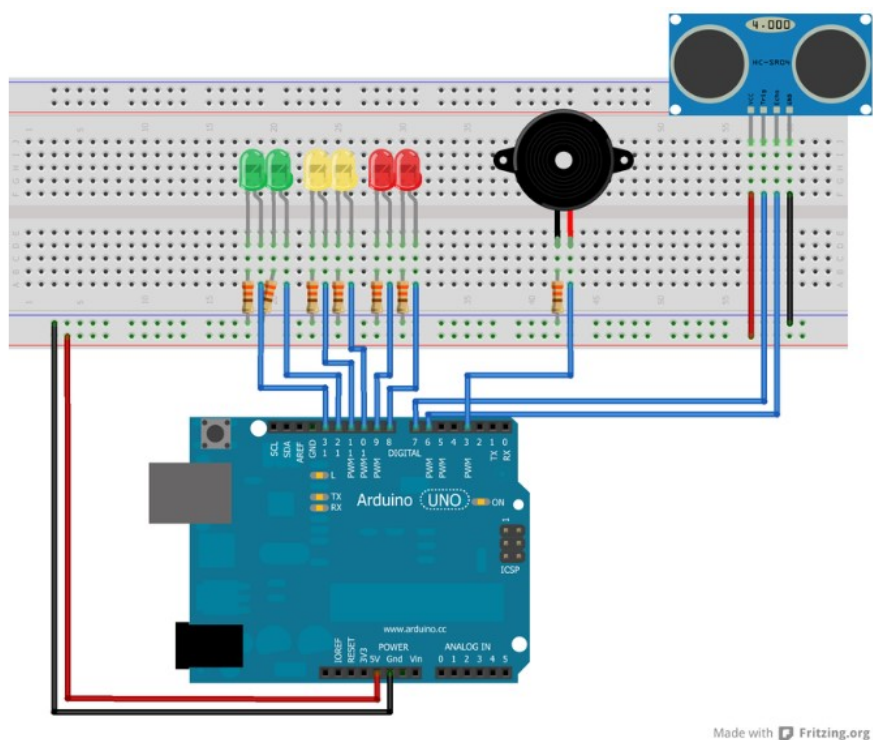
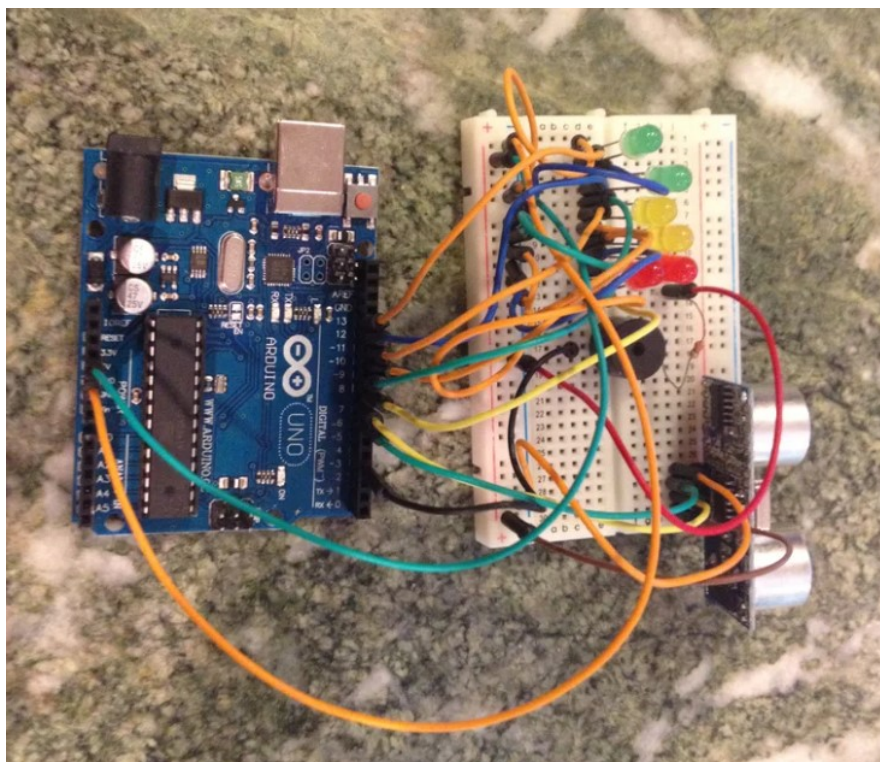


Foto realizace



Kód programu

```
// Definice pinů pro ultrazvukový senzor
const int pin_echo = 6;
const int pin_trig = 7;

const int buzzer_pin = 3;

const int led_green1 = 13;
const int led_green2 = 12;
const int led_yellow1 = 11;
const int led_yellow2 = 10;
const int led_red1 = 9;
const int led_red2 = 8;

const int ledPins[] = { led_green1, led_green2, led_yellow1,
led_yellow2, led_red1, led_red2 }; // Pole s piny pro LED diody
const int vzdalenosti[] = { 30, 25, 20, 15, 10, 5 };
                                // Hodnoty vzdáleností pro každou LED
const int tony[] = { 250, 260, 270, 280, 290, 300 };
                                // Odpovídající hodnoty zvuku pro každou
vzdálenost

// Proměnná pro ukládání změřené vzdálenosti
unsigned long vzdalenost;

// Proměnná pro ukládání tónu
int ton = 0;

void setup() {
    // nastavení pinů pro ultrazvukové čidlo
    pinMode(pin_trig, OUTPUT);
    pinMode(pin_echo, INPUT);

    // Nastavení pinů pro ledky
    for (int i = 0; i < 6; i++) {
        pinMode(ledPins[i], OUTPUT);
    }

    // inicializace sériového portu
    Serial.begin(9600);
}
```

```

void loop() {
    // Volání funkce pro výpočet vzdálenosti
    vzdalenost = zmer_vzdalenost(pin_trig, pin_echo);

    // Výpis vzdálenosti na sériový monitor
    Serial.print("Vzdalenost: ");
    Serial.print(vzdalenost);
    Serial.println(" cm");

    // Rozsvícení LEDek a nastavení tónu podle vzdálenosti
    for (int i = 0; i < 6; i++) {
        if (vzdalenost <= vzdalenosti[i]) {
            digitalWrite(ledPins[i], HIGH);
            ton = tony[i];
        } else {
            digitalWrite(ledPins[i], LOW);
        }
    }

    if (vzdalenost > 30 || vzdalenost <= 0) {
        noTone(buzzer_pin);
    } else {
        tone(buzzer_pin, ton);
    }
    delay(100);
}

// Funkce pro výpočet vzdálenosti
unsigned long zmer_vzdalenost(int trig, int echo) {
    unsigned long vzdalenost_cidlo, cas_cidlo;
    // Vyšle ultrazvukové vlny pomocí TRIG pinu
    digitalWrite(trig, HIGH);
    delay(1);
    digitalWrite(trig, LOW);
    cas_cidlo = pulseIn(echo, HIGH);
    // Převedení času na vzdálenost
    vzdalenost_cidlo = (cas_cidlo / 2) / 29.1;
    return vzdalenost_cidlo;
}

```

Závěr

Povedlo se mi implementovat všechny funkce požadované zadáním. Nejpracnější bylo vyladit cyklus, který podle vzdálenosti nastavuje jednotlivé LEDky. Dále jsem řešil problém, že každé druhé měření ukazovalo nulovou vzdálenost. Pomohlo doporučení v datasheetu ultrazvukového senzoru, že mezi jednotlivými měřeními by mělo být alespoň 60ms pauza. Po přidání 100ms zpoždění už měření fungovalo správně.

Použité zdroje a knihovny

Knihovna New Ping: <https://bitbucket.org/teckel12/arduino-new-ping/wiki/Home>

HC-SR04 tutorial - <https://howtomechatronics.com/tutorials/arduino/ultrasonic-sensor-hc-sr04/>

HC-SR04 datasheet -

<https://www.smartprototyping.com/image/data/2020/09/102042%20HC-SR04%20Ultrasonic%20Distance%20Sensor/HCSR04%20Datasheet.pdf>