4. Displej a senzory

Obsah

Použité součástky a moduly

V dnešní úloze budete potřebovat displej a 3 senzory. Pokud nemáte 3 senzory, poraďte se se cvičícím, jak chybějící senzory nahradit.

LCD displej 1602

Znakový LCD displej 1602 umí zobrazit až 16 znaků na 2 řádcích. Ovládá se paralelním rozhraním, v 8b nebo 4b režimu. Pokud máte k displeji převodník i2c (../12/index.html#chap-i2c) , stačí vám na ovládání displeje 2 signály, jinak budete potřeboval alespoň 6 pinů (4 pro data + 2 řídící).

Prostudujte si dokumentaci ke knihovně LiquidCrystal_i2c (https://www.arduino.cc/reference/en/libraries/liquidcrystal-i2c/) , případně LiquidCrystal (https://www.arduino.cc/en/Reference/LiquidCrystal) .

Teploměr

Pokud máte jeden z doporučených teplotních senzorů, čtěte níže. Pokud máte jiný, nastudujte si dokumentaci, jak se používá, případně si najděte knihovny pro práci se senzorem.

Digitální senzor DHT11

Senzor DHT11 (a DHT22) kombinuje vlhkoměr a teploměr. S Arduinem komunikuje vlastním protokolem po jednom datovém vodiči. My pro čtení dat ze senzoru použijeme knihovnu DHT (https://www.arduino.cc/reference/en/libraries/dht-sensor-library/) .

Analogový senzor

LM335 (cvičení)

Senzor LM335 budeme používat v nejjednodušším zapojení, jako Zenerovu diodu tedy sériově s rezistorem pro omezení proudu. Kalibrační výstup necháme nezapojený.

Napětí na senzoru závisí lineárně na teplotě. Z výroby je kalibrován tak, že napětí přímo odpovídá teplotě v Kelvinech: $T = \frac{U_t}{10\,\mathrm{mV}}\mathrm{K}$ Jinými slovy, pokud je na senzoru 3.15 V, odpovídá to teplotě 315 K (42 ˚C).

Senzor pracuje stabilně pro prouv rozmezí mezi 400 µA a 5 mA. Zvolíme tedy rezistor tak, aby rezistorem protékaly přibližně 2 mA. $R = \frac{U_{cc}-U_t}{I} = \frac{(5-3)\mathrm{V}}{2\mathrm{mA}} = 1\mathrm{k}\Omega$

Další podrobnosti lze najít v dokumentaci: https://www.ti.com/lit/ds/symlink/lm335.pdf

TMP36 (simlátor)

Senzor TMP36 pracuje s napájecím napětím 2.7–5.5 V, jeho výstupem je napětí 0.1–2.0 V, v závislosti na teplotě. Toto napětí je pak třeba přepočítat na teplotu.

Vzorec pro výpočet teploty

$T = \frac{V_o - 500}{10}$

V_o v mV, T v °C

Vzorec je převzat z https://learn.adafruit.com/tmp36-temperature-sensor.

LM19

Senzor LM19 pracuje s napájecím napětím 2.5–5.5 V, jeho výstupem je napětí 0–2.5 V, v závislosti na teplotě. Toto napětí je pak třeba přepočítat na teplotu.

Vzorec pro výpočet teploty

$T = -1481.96 + \sqrt{2.1962 \cdot 10^6 + \frac{1.8639 - V_o}{3.88 \cdot 10^{-6}}}$

V_o ve V, T v °C

Vzorec je převzat z dokumentace k LM19 (https://www.ti.com/product/LM19) .

Světelný senzor

Fotorezistor je odporová součástka, která mění svůj odpor v závislosti na množství dopadajícího světla. Při sériovém spojení s pevným rezistorem (nebo s potenciometrem) vznikne odporový dělič. Jeho dělící poměr (a měřené napětí) závisí na intenzitě osvětlení. Rezistor volte tak, aby rozsah výstupního napětí byl co největší.

1.úloha - Zobrazení výstupu senzorů na displej (7 b)

Zobrazte na displej hodnotu teploměru, mikrofonu a světelného senzoru (nebo jiných senzorů, které máte k dispozici). Vytvořte menu, kde bude možné si zvolit jeden z těchto senzorů a vypsat jeho výstup na displej. Menu se bude ovládat pomocí 4 tlačítek `SW1` až `SW4`.

PŘÍKLAD MAPOVÁNÍ TLAČÍTEK:

- `SW1` : pohyb dolu
- `SW2` : vybrat senzor
- `SW3` : zpět do menu
- `SW4` : pohyb nahoru

Domovská obrazovka

Menu neboli domovská obrazovka slouží jako rozcestník pro výpis jednotlivých senzorů. Může vypadat například takto:

Příklad 1. Domovská obrazovka:

```
> 1. Temperature
  2. Microphone

// druhá „obrazovka“

> 3. Light
```

Pohyb po menu se zobrazí pomocí znaku `'>'` před prvkem seznamu. (Zde se tedy aktuálně nacházíte na položce Temperature.) Dejte pozor, aby displej při pohybu po menu neblikal (tj. nemažte při posuvu znaku `'>'` celou obrazovku, ale pouze tento znak).

Pro každý senzor zvolte vhodný způsob zobrazení jeho hodnoty.

Příklady zobrazení některých senzorů:

Teploměr

Zvolením položky teploměr v menu, přejdete na obrazovku, kde se bude průběžně zobrazovat teplota (aktualizovaná každé 2 sekundy) ve stupních Celsia.

Pro obnovu dat po 2 sekundách, využijte funkce `millis()` z 1. cvičení.

Příklad 2. Obrazovka teploměru

```
TEMPERATURE
```

```
21 C
```

Pro převedení čísla na text můžete použít datový typ `String` (https://www.arduino.cc/reference/en/language/variables/data-types/stringobject/) a jeho metodu `c_str()`.

Mikrofon

Zvolením položky mikrofon v menu, přejdete na obrazovku, kde se bude vykreslovat křivka podle momentálního zvuku z mikrofonu. Posuňte křivku na ose y lehce nahoru tak, aby byla vidět i v klidovém stavu (viz obrázek).

Příklad 3. Obrazovka mikrofonu

```
MICROPHONE
======    |
```

Zde je aktuální hodnota reprezentována znaky `=` a znak `|` označuje maximální hodnotu (za posledních x sekund).

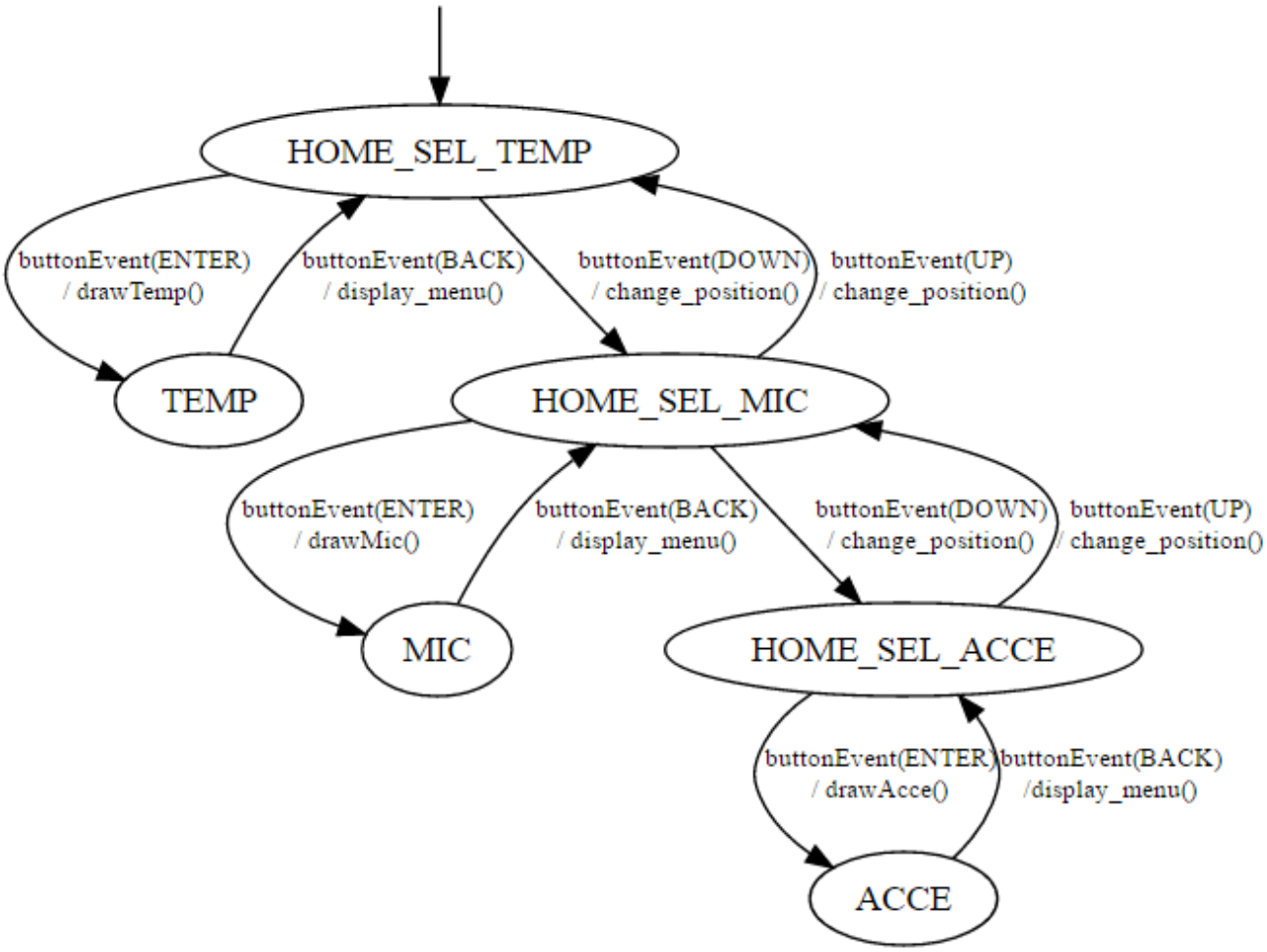
Akcelerometr

Po zvolení položky akcelerometr v menu, zobrazíte na displeji kolečko, které se bude pohybovat v závislosti na aktuálním náklonu senzoru.

Příklad 4. Obrazovka akcelerometru

```
       v
----o----------
```

Implementace pomocí konečného automatu



Obrázek 1. Graf přechodů stavového automatu, který tvoří kostru programu

Můžete použít následující základní strukturu programu.

Ukázka 1. Kostra programu:

```
enum states {
  HOME_SEL_TEMP, HOME_SEL_MIC, HOME_SEL_ACCE, TEMP, MIC, ACCE
};
enum states STATE, NEXT_STATE;

void setup() {
 // Proveďte inicializaci displeje
  display_menu();                                  // zobrazení domovské obrazovky
  STATE = HOME_SEL_TEMP;
}

void loop()
{
  switch (STATE)
   {
      case HOME_SEL_TEMP:                          // MENU - TEMPERATURE
          if(buttonEvent(DOWN))                    // je stisknuto tlačítko 'pohyb dolu'?
          {
```

```
      change_position();                // změna domovské obrazovky
      NEXT_STATE = HOME_SEL_MIC;
    }
    else if (buttonEvent(ENTER))        // je stisknuto tlačítko 'vybrat senzor'?
    {
      NEXT_STATE = TEMP;
    }
    break;

case HOME_SEL_MIC:                      // MENU - MICROPHONE
    if (buttonEvent(DOWN))              // je stisknuto tlačítko 'pohyb dolu'?
    {
      change_position();                // změna domovské obrazovky
      NEXT_STATE = HOME_SEL_ACCE;
    }
    else if (buttonEvent(UP))           // je stisknuto tlačítko 'pohyb nahoru'?
    {
      change_position();                // změna domovské obrazovky
      NEXT_STATE = HOME_SEL_TEMP;
    }
    else if (buttonEvent(ENTER))        // je stisknuto tlačítko 'vybrat senzor'?
    {
      NEXT_STATE = MIC;
    }
    break;

case HOME_SEL_ACCE:                     // MENU - ACCELEROMETER
    if (buttonEvent(UP))                // je stisknuto tlačítko 'pohyb nahoru'?
    {
      change_position();                // změna domovské obrazovky
      NEXT_STATE = HOME_SEL_MIC;
    }
    else if (buttonEvent(ENTER))        // je stisknuto tlačítko 'vybrat senzor'?
    {
      NEXT_STATE = ACCE;
    }
    break;

case TEMP:                              // DRAW TEMPERATURE
    if(buttonEvent(BACK))               // je stisknuto tlačítko 'zpět do menu'?
    {
      display_menu();                   // zobrazení domovské obrazovky
      NEXT_STATE = HOME_SEL_TEMP;
    }
    else
    {
      draw_temp();                      // funkce draw_temp vykreslí aktuální teplotu každé 2 sekur
    }
```

```
            break;

        case MIC:                               // DRAW MICROPHONE
            if(buttonEvent(BACK))               // je stisknuto tlačítko 'zpět do menu'?
            {
              display_menu();                   // zobrazení domovské obrazovky
              NEXT_STATE = HOME_SEL_MIC;
            }
            else
            {
              draw_mic();                       // funkce draw_mic vykreslí křivku podle momentálního zvuku
            }
            break;

        case ACCE:                              //  DRAW ACCELEROMETER
            if(buttonEvent(BACK))               // je stisknuto tlačítko 'zpět do menu'?
            {
              display_menu();                   // zobrazení domovské obrazovky
              NEXT_STATE = HOME_SEL_ACCE;
            }
            else
            {
              draw_acce();                     // funkce draw_acce zobrazí kolečko, které se bude pohybovat
            }
            break;
    }
    STATE = NEXT_STATE;
}
```

Použijte následující funkci `buttonEvent(int button)` ve svém programu. Tuto funkci využijete při implementaci pohybu po domovské obrazovce. Funkce vrací true, jestliže bylo zadané tlačítko (parametr button) stisknuto a puštěno, případně false pokud stisknuto nebylo nebo naopak nebylo ještě puštěno.

> V simulátoru Tinkercad je chyba, která se projevuje v následující funkci `buttonEvent()`. Kód jsme upravili, aby se chyba neprojevila.
>
> Chyba se projevuje chybovou hláškou `error: expected unqualified-id before 'else'` na úplně nesouvisejícím místě v kódu.
>
> Pokud už máte zkopírovanou starší verzi této funkce (nebo jste to v kódu napsali sami), tak stačí smazat whitespace kolem `else` - `else` mějte na stejném řádku jako předcházející `}`.

Ukázka 2. Funkce pro detekci stisku tlačítka

```
#define DOWN  SW1
```

```
#define UP    SW4
#define ENTER SW2
#define BACK  SW3

byte buttonFlag = 0;

bool buttonEvent(int button)
{
  switch(button)
  {
    case UP:
     if (digitalRead(UP) == LOW)
     {
       buttonFlag |= 1;
     } else if (buttonFlag & 1)
     {
       buttonFlag ^= 1;
       return true;
     }
     break;

    case DOWN:
     if (digitalRead(DOWN) == LOW)
     {
       buttonFlag |= 2;
     } else if (buttonFlag & 2)
     {
       buttonFlag ^= 2;
       return true;
     }
     break;

    case BACK:
     if (digitalRead(BACK) == LOW)
     {
       buttonFlag |= 4;
     } else if (buttonFlag & 4)
     {
       buttonFlag ^= 4;
       return true;
     }
     break;

    case ENTER:
     if (digitalRead(ENTER) == LOW)
     {
       buttonFlag |= 8;
     } else if (buttonFlag & 8)
```

```
      {
        buttonFlag ^= 8;
        return true;
      }
    }
    return false;
}
```
