

5. SD karta

Obsah

1. [Perzistentní paměti](#)
 1. [EEPROM](#)
 2. [SD karta](#)
2. [Úlohy](#)
 1. [1. úloha - Zobrazení kořenového adresáře \(4 b\)](#)
 2. [2. úloha - Zobrazení obrázku z SD karty \(2 b\)](#)
 3. [3. úloha - Secret Code \(4 b\)](#)

Na dnešním cvičení budete pracovat s přípravkem Arduino Esplora. Na rozdíl od jiných přípravků Arduino jsou periferie na Esplore pevně připájené. Není tedy potřeba nic zapojovat.

Periferie na Esplore budete ovládat pomocí knihovny `Esplora.h`, prostudujte si dostupné funkce v [dokumentaci](https://docs.arduino.cc/retired/archived-libraries/EsploraLibrary) (<https://docs.arduino.cc/retired/archived-libraries/EsploraLibrary>).

Budete dále pracovat s grafickým displejem a knihovnou [TFT](https://www.arduino.cc/en/Reference/TFTLibrary) (<https://www.arduino.cc/en/Reference/TFTLibrary>).

Perzistentní paměti

V 5. cvičení se seznámíte s problematikou tzv. perzistentních pamětí (EEPROM, SD karta). Perzistentní paměti umožňují, aby uložená data byla dostupná i po přerušení napájení. Tento typ paměti nazýváme také nevolatilní.

EEPROM

Mikroprocesor přípravku Arduino Esplora obsahuje 1024 B paměti EEPROM (Electrically Erasable Programmable Read-Only Memory).

Životnost EEPROM nepřesahuje 100 000 cyklů zápisů/mazání. Zápis na EEPROM trvá cca 3,3 ms.

Součástí platformy Arduino je knihovna EEPROM, usnadňující práci s touto pamětí. Podrobnější informace naleznete na: <https://www.arduino.cc/en/Reference/EEPROM>.

SD karta

Prostudujte si knihovnu funkcí SD karty: <https://www.arduino.cc/en/Reference/SD>.

Úlohy

1. úloha - Zobrazení kořenového adresáře (4 b)

Vaším prvním úkolem je vypsát na TFT displej obsah kořenového adresáře SD karty.

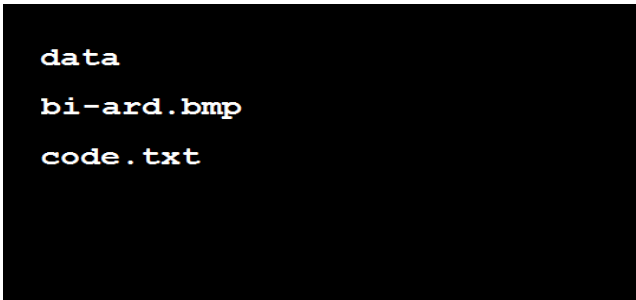
Nejprve inicializujte TFT LCD displej stejně jako v minulém cvičení (nezapomeňte k vašemu programu připojit správné knihovny pro práci s displejem). Přidejte k programu také knihovnu SD karty. Inicializujte SD kartu pomocí funkce `SD.begin(cspin)`, jejímž parametrem je číslo CS (chip select) pinu. Tento pin je nezbytný pro komunikaci mikrokontroléru s SD kartou, viz [SPI komunikace \(https://cs.wikipedia.org/wiki/Serial_Peripheral_Interface\)](https://cs.wikipedia.org/wiki/Serial_Peripheral_Interface). Přidejte proto do programu řádek:

```
#define SD_CS 8 // Chip select line for SD card in Esploa
```

Makro `SD_CS` poté zadejte jako parametr funkci `begin()`. Pokud funkce `begin()` vrátí `false`, vypište na displej „No SD card inserted!“ a ukončete program.

Dále nastavte barvu pozadí na černou a barvu textu na bílou. Pomocí funkce `SD.open()` otevřete kořenový adresář a vypište jeho obsah na displej. Pro vypsání souborů v adresáři využijte funkci `SD.openNextFile()`.

Vaše úloha by po dokončení měla vypadat na displeji takto:



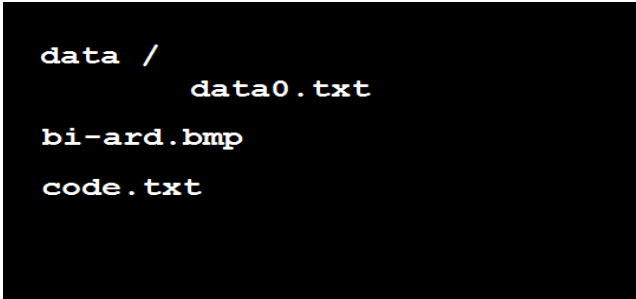
```
data
bi-ard.bmp
code.txt
```

Obrázek 1. Příklad výpisu kořenového adresáře

Bonus - Zobrazení adresářové struktury SD karty (+ 1 b)

Vypište na displej celou adresářovou strukturu SD karty. Pro vypsání obsahu jednotlivých adresářů doporučujeme použít rekurzi.

Vaše úloha by po dokončení měla vypadat na displeji takto:



```
data /
      data0.txt
bi-ard.bmp
code.txt
```

Obrázek 2. Příklad výpisu adresářové struktury



Neměňte obsah SD karty. (Na dalším cvičení vás nebudou mít rádi.)

2. úloha - Zobrazení obrázku z SD karty (2 b)

Nejprve inicializujte TFT LCD displej. Nastavte pozadí displeje na bílou barvu. Poté inicializujte SD kartu pomocí funkce `SD.begin()`. Pokud funkce `begin()` vrátí `false`, vypište na displej „No SD card inserted!“ a ukončete program.

Zkontrolujte, zda se v kořenovém adresáři SD karty nachází obrázek s názvem `bi-ard.bmp`. Využijte k tomu funkce `exists()` z knihovny SD. Pokud funkce `exists()` vrátí `false`, vypište na displej „bi-ard.bmp missing!“ a ukončete program. Obsah SD karty nijak neměňte! Pokud obrázek na kartě naleznete, nahrajte ho pomocí funkce `loadImage()` a zkontrolujte, zda se jedná o korektní soubor funkcí `isValid()`. Pokud je obrázek validní, zablikejte zelenou RGB LED, pokud ne, pak zablikejte červenou.

Nyní zobrazte obrázek na displej do levého horního rohu.

Bonus - Posouvání obrázku po displeji (+ 1 b)

Posouvajte obrázek po displeji v závislosti na pohybu joysticku. Při stisknutí tlačítka `SWITCH_1`, uložte souřadnice polohy `x` a `y` do paměti EEPROM. Po resetování přípravku zobrazte obrázek v poloze určené souřadnicemi uloženými v EEPROM.

Nejprve je třeba nahrát do EEPROM prvotní souřadnice, kde `x = 0` a `y = 0`. Vytvořte proto **zvlášť nový sketch**, kde pouze ve funkci `setup` dvakrát zavoláte funkci `EEPROM.update()`, například následovně:

```
#include <EEPROM.h>

void setup() {
  EEPROM.update(0,0);    // Ukládám hodnotu 0 na adresu 0 v paměti EEPROM.
  EEPROM.update(1,0);    // Ukládám hodnotu 0 na adresu 1 v paměti EEPROM.
}

void loop() {
}
```

Takto do paměti EEPROM nahrajete hodnotu 0 na adresy 0 a 1 (pouze v případě, že se na těchto adresách stejné hodnoty už nevyskytují, v tom případě nedochází k přepsání). Nahrajte kód do přípravku, zavřete tento sketch a vraťte se k původnímu programu z 2. úlohy. Přidejte detekci pohybu joysticku, tj. uložte aktuální hodnoty souřadnic `x` a `y`. Tyto hodnoty je potřeba namapovat na velikost displeje. Použijte k tomu funkci [map](https://www.arduino.cc/en/Reference/Map) (<https://www.arduino.cc/en/Reference/Map>). Velikost displeje zjistíte pomocí funkcí `EsploraTFT.width()` a `EsploraTFT.height()`. Poté zobrazte na displej obrázek na nových souřadnicích. Vykreslujte obrázek v delším časovém intervalu (např. 500 ms).

V případě, že došlo ke stisknutí tlačítka `SWITCH_1`, použijte funkci `EEPROM.update()` k uložení aktuálních souřadnic. Aby se po vypnutí přípravku zobrazil obrázek na uložených souřadnicích, zavolejte ve funkci `setup()`, dvakrát funkci `EEPROM.read()` a přečtete tak uložené hodnoty souřadnic `x` a `y`. Návratová hodnota funkce `read()` je číslo uložené v paměti EEPROM na adrese, kterou zadáte jako parametr. Podle předchozího příkladu bychom tedy pro získání hodnoty souřadnice `x` volali `x = EEPROM.read(0)`.



Životnost paměti EEPROM je omezena na 100 000 zápisů/mazání. Vyhněte se proto příliš častému zápisu do paměti. Pokud nemusíte, použijte funkci [EEPROM.update](https://www.arduino.cc/en/Reference/EEPROMUpdate) (<https://www.arduino.cc/en/Reference/EEPROMUpdate>) místo [EEPROM.write](https://www.arduino.cc/en/Reference/EEPROMWrite) (<https://www.arduino.cc/en/Reference/EEPROMWrite>).

3. úloha - Secret Code (4 b)

Vaší další úlohou bude nasimulovat fungování trezoru. Pomocí tlačítek na Esploře zadáte posloupnost čísel (neboli bezpečnostní heslo) a pokud je tento vybraný kód správný, trezor se vám otevře, jinak zůstane zamčený.

Na začátku je třeba trezor uzamknout. To provedete tak, že stisknete tlačítko joysticku, rozsvítíte červenou RGB LED a na LCD displej vypíšete:

```
*** Locked! ***
```

Enter code:

Nyní následuje zadání hesla. Nejdříve na SD kartě otevřete soubor `code.txt`, kde najdete uložené bezpečnostní heslo. Toto heslo poté načtete do svého programu například tomto formátu:

```
int secret_code[4]    // secret code read from SD
```

Pole `secret_code[]` tedy bude obsahovat správný bezpečnostní kód (definovaný souborem na SD) a pořadí stisknutých tlačítek se musí shodovat s číslicemi, které jsou v tomto poli uloženy. Po zadání každé vstupní číslice pomocí tlačítka, vypište na displej hvězdičku (*). Načtete všechny čtyři číslice a *teprve poté*, pokud byla nějaká číslice chybná, proveďte výpis ve tvaru:

```
Wrong secret code!
```

```
*** Locked! ***
```

Enter code:

V případě, že kód byl zadán bez chyby, vypište:

```
*** Unlocked! ***
```

a rozsvítte zelenou RGB LED. Takto signalizujete otevření trezoru.