

1. Robot

S využitím IO karty sestavte v jazyku C# program pro ovládání modelu robota. Dle svého uvážení zvolte buď prostředí konzolové aplikace nebo grafické prostředí.

Výsledné řešení musí obsahovat následující funkcionalitu:

- Automatické nastavení výchozí polohy všech pohonů
- Manuální ovládání robota prostřednictvím ovladače s ukládáním provedeného pohybu do souboru v libovolné podobě (zamyslete se nad efektivností zvoleného formátu)
- Zopakování manuálně naučeného pohybu uloženého v souboru
- Průběžná a použitelná indikace aktuální operace a stavu zařízení na monitoru

2. Generátor průběhů

S využitím vývojového kitu AVR a dvou paralelních DAC převodníků sestavte v jazyku C program fungující jako jednoduchý dvoukanálový funkční generátor ovládaný ze sériové linky. Pro její obsluhu použijte knihovnu, kterou máte k dispozici v rámci ukázkových kódů. Ověření parametrů generovaného průběhu provádějte osciloskopem.

Výsledné řešení musí obsahovat následující funkcionalitu:

- Nezávislé nastavení parametrů pro každý kanál v reálných jednotkách (Hz, mV)
- Podpora tří různých průběhů na kanál. Vyžadován sinus a dva další dle vlastního výběru
- Generování průběhu v reálném čase pro oba kanály současně
- Nastavení parametrů průběhu pomocí textových příkazů z terminálu připojeného sériovou linkou (např. CH1:TYPE:SIN, CH2:FREQ:12, CH1:AMP:1500)

3. Zpracování DCF časové informace

S využitím vývojového kitu AVR a přijímače DCF časové informace (buď v reálné nebo virtualizované podobě) sestavte v jazyku C program implementující čtení a dekodování časové informace. Tu zobrazte do terminálu připojeného na sériové lince procesoru. Pro její obsluhu použijte knihovnu, kterou máte k dispozici v rámci ukázkových kódů.

Výsledné řešení musí obsahovat následující funkcionalitu:

- Spolehlivá synchronizace a vyhodnocení jednotlivých časových značek (bitů) z přijímače
- Uložení přijaté informace do vhodně zvolené datové struktury umožňující její následné vyhodnocení
- Dekodování přijaté informace do čitelné podoby, detekce chyb a částečné vyhodnocení nekompletní informace
- Zobrazení dekodované informace, příp. její části, do terminálu připojeného sériovou linkou.

4. Přístupový systém s RFID

S využitím periferie čteček RFID čipů připojené pomocí USB / sériového rozhraní sestavte aplikaci, která bude na základě informací uložených v SQLite databázi vyhodnocovat oprávněnost požadavku na přístup z daného místa.

Výsledné řešení musí obsahovat následující funkcionalitu:

- Obsluha dvou čteček připojených na periférii pomocí RS485 rozhraní
- Identifikace čtečky a způsobu načtení informace (načtení čipu, manuální zadání kódu na klávesnici). Simulujte tři místa události (klávesnice u vstupu, čip v místě A, čip v místě B)
- Identifikace osoby a vyhodnocení oprávněnosti přístupu na základě *místa* události, *aktuálního času* a *dne v týdnu*
- *Uživatelsky použitelné* rozhraní pro nastavení údajů do databáze (vlastník, členství ve skupinách, nastavení přístupových oprávnění a pravidel pro přístup)
- Zobrazení výsledku vyhodnocení žádosti a logování všech událostí do tabulky databáze ve formátu umožňujícím efektivní prohlédávání na úrovni SQL dotazu (zamyslete se jak by mohlo vypadat např. dohledání vstupů do konkrétního prostoru)

5. Mikrovlnná trouba

S využitím IO karty sestavte v jazyku C# program pro ovládání modelu mikrovlnné trouby. Dle svého uvážení zvolte buď prostředí konzolové aplikace nebo grafické prostředí.

Výsledné řešení musí obsahovat následující funkcionalitu:

- Nastavení času ohřevu (alespoň po desítkách sekund) a výkonu ohřevu ve 2 úrovních
- Manuální spuštění cyklu ohřevu s možností kdykoliv jej přerušit, otevřít dvířka a dále pokračovat nebo zrušit
- Zvukové znamení a zhasnutí po skončení ohřevu, pomalou rotaci talíře během cyklu ohřevu
- Kontrolu bezpečnosti provozu (nemožnost spustit s otevřením dvířek, automatické přerušení ohřevu při porušení bezpečnosti)

6. Světelný řádek

S využitím vývojového kitu AVR a periferie sériově ovládaného světelného řádku 96x16 bodů sestavte v jazyku C program fungující jako jednoduchý reklamní poutač

Výsledné řešení musí obsahovat následující funkcionalitu:

- Softwarovou implementaci SPI komunikace s řadičem displejů MAX7219 s vhodným aplikačním rozhraním
- Podporu zobrazení alespoň tří nezávislých textů napevno uložených v programu a jejich cyklické zobrazování
- Možnost jednoduché změny zobrazovaného textu na úrovni kódu (např. úpravou řetězce / pole s daty)
- Podporu zobrazení alespoň jedné sady znaků (malá/velká písmena) a čísel. Celý font nemusíte počítat ručně, využijte libovolný z dostupných (nebo svých) nástrojů. Musíte však vědět jak formát znaku vypadá (a ideálně si nepřidělovat zbytečnou práci v kódu několikerou konverzí připravených dat fontu).

7. Výtah

S využitím IO karty sestavte v jazyku C# program pro ovládání modelu výtahu. Dle svého uvážení zvolte buď prostředí konzolové aplikace nebo grafické prostředí.

Výsledné řešení musí obsahovat následující funkcionalitu:

- Spolehlivá inicializace výchozího stavu zařízení
- Ovládání výtahu v rozsahu min. čtyř pater z kabinových a patrových tlačítek
- Zpracování informací jednotlivých čidel modelu (patro, podlahový spínač apod.) a reakce podobná s reálným zařízením spolu s indikací stavu výtahu pomocí patrové indikace
- Implementace alespoň základní bezpečnosti provozu během pohybu kabiny

8. Automatizované zpracování dat – systém prodeje a výdeje stavebnin

S využitím periferie čtečky čárového kódu sestavte C# aplikaci, která ve spojení s SQLite databází bude fungovat jako prodejní a výdejní systém ve stavebninách. Pro potřeby prodeje pracujte s daty definujícími prodávané zboží, jeho dostupnost (počet ks, kg apod.) a jednotkovou cenu. Editor pro správu těchto dat není součástí řešení. Při prodeji podle požadavků zákazníka obsluha načítá kódy jednotlivých artiklů (např. z vytištěného seznamu) a systém automaticky kontroluje dostupnost. Po ukončení prodeje a zaplacení požadované částky (uvažujte pouze hotovostní platbu) je zákazníkovi vytvořena výdejka („faktura“) obsahující identifikaci obsluhy, jednotlivé zakoupené položky, jednotkové ceny a celkovou cenu spolu s dalšími náležitostmi. Tento doklad si obsluha skladu načte pomocí jeho unikátního kódu a zobrazí si tak položky určené k výdeji pro daného zákazníka. Všechny informace o provedených operacích se zároveň ukládají do databáze ve formátu vhodném pro následný audit (např. zjištění průměrného objemu prodeje artiklu za období, tvorba cenových akcí nebo sledování průměrné doby na vyskladnění zakázky).

Výsledné řešení musí obsahovat následující funkcionalitu:

- Jednoduché zadání nákupu s využitím čtečky čárového kódu a *numerického bloku klávesnice bez myši*. Zobrazujte průběžnou cenu nákupu a částku pro vrácení obsluhou při platbě. Zvažte efektivitu zadávání např. palety cihel vs. načítání kódu pro každou cihlu na paletě.
- Korektní reakci na chybný vstup (neznámá nebo nedostupná skladová položka, neplatný kód výdejky apod.)
- Výdejky ukládané v databázi spolu s uložením do vybraného typu souboru vč. kódu výdejky (alespoň jako obrázek).
- Minimální rozhraní pro výdej zakázky zobrazující seznam položek k vyskladnění. Skladník neprovádí další interakci se systémem, předpokládá se pouze zobrazení a kompletní výdej každé zakoupené položky.
- Hodnoceno je mj. uživatelské prostředí a použitelnost aplikace, odolnost proti chybám a dodržení alespoň základních principů pro korektní návrh struktury a vazeb tabulek databáze