

1. Robot

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

Výsledné řešení musí obsahovat následující funkcionalitu:

- Automatické nastavení výchozí polohy všech pohonů (pozor na implementaci výchozí polohy základny)
- Manuální ovládání robota prostřednictvím herního ovladače s ukládáním provedeného pohybu do souboru v libovolné podobě. Zamyslete se nad efektivitou práce s tímto souborem a jeho velikostí.
- Zopakování manuálně naučeného pohybu načteného ze 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. Ověření parametrů generovaného průběhu provádějte osciloskopem.

Výsledné řešení musí obsahovat následující funkcionalitu:

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

3. RTC – hodiny reálného času

S využitím vývojového kitu AVR a periferie čipu reálného času DS3231 sestavte v jazyku C program implementující komunikaci s RTC čipem pomocí I2C sběrnice.

Výsledné řešení musí obsahovat následující funkcionalitu:

- Implementace I2C komunikace na úrovni softwaru nebo využitím hardwarové jednotky I2C v procesoru (s úplným pochopením principů, nikoliv copy & paste nalezeným fragmentem kódu)
- Čtení informace o aktuálním datu a čase s využitím vnitřních registrů obvodu dle katalogového listu
- Zobrazení informace přečtené z RTC na terminál připojený sériovou linkou k vývojovému kitu (s použitím komunikační knihovny z příkladů kódu)
- Ověření zápisu do obvodu pomocí nastavení libovolného pevného data nebo času (nastavte jako aktuální např. datum prvního cvičení úlohy)

4. Přístupový systém s RFID

S využitím periferie čteček RFID čipů připojené pomocí USB / sériového rozhraní sestavte aplikaci, která bude fungovat jako přístupový systém sportovního zařízení s evidencí a automatickým zpoplatněním stráveného času. Veškerá data jsou uložena v SQLite databázi, přístup je umožněn oprávněným uživatelům s kontrolou času a dne v týdnu (podle aktivního předplatného), na jejichž účtu je předplacený kredit čerpaný při vstupu.

Výsledné řešení musí obsahovat následující funkcionalitu:

- Obsluha dvou čteček připojených na modelu pomocí RS485 rozhraní
- Malá čtečka po přihlášení „správcovským“ čipem aktivuje rozhraní pro správu zákazníků, kde lze provádět základní operace pro správu jejich předplatného.
- Velká čtečka umožňuje vstup zákazníka buď načtením registrované karty nebo zadáním osobního pinu (zapomenutý čip). Se vstupem (pokud je v danou dobu povolen) začíná účtování vstupu z dostupného kreditu, s odchodem (odhlášením stejným čipem nebo kódem) se aktualizuje kredit zákazníka spolu s informací o čerpání.
- Uživatelsky použitelné rozhraní pro nastavení údajů do databáze (editace zákazníka a jeho osobního kódu, úroveň předplatného, cenu hodinového vstupu pro jednotlivé úrovně předplatného, doby povolených vstupů pro jednotlivé typy předplatných – např. všední den, večerní zlevněný vstup, víkendový vstup atd.)
- Zobrazení výsledku při požadavku na vstup (uvažujte pouze informační displej s možnými stavy vyhodnocení pro zákazníka) spolu s logováním všech operací (např. pro účely řešení reklamací) do tabulky (tabulek) databáze ve formátu umožňujícím prohledávání na úrovni SQL dotazu. Zamyslete se, jak by mohlo např. vypadat dohledání všech změn kreditu konkrétního zákazníka, nebo zjištění zákazníků přítomných v zařízení v zadaný datum a čas.

5. Zabezpečená výstavní skříň (MVT)

S využitím IO karty sestavte v jazyku C# program pro ovládání modelu mikrovlnné trouby ve funkci bezpečnostní výstavní schránky. Dle vlastního uvážení volte buď prostředí konzolové aplikace nebo grafické prostředí.

Výsledné řešení musí obsahovat následující funkcionalitu:

- V uzavřeném stavu aktivujte osvětlení na 50% maxima a pomalé otáčení talíře pro prezentaci vystaveného předmětu
- Po zadání správného kódu zastavení otáčení, rozsvícení na 100% a otevření dvířek pro úpravu expozice. V otevřeném stavu je možné změnit bezpečnostní kód.
- Zvukové znamení pokud jsou dvířka otevřená déle než 1 minutu a v případě zadání chybného kódu 3x za sebou
- Perzistentní (trvalé) bezpečné uložení nastaveného kódu tak, aby nebyl ovlivněn mezi spuštěními aplikace.

6. Světelný řádek

S využitím vývojového kitu AVR a periferie sériově ovládaného světelného řádku 96x16 bodů s řadiči MAX7219 sestavte v jazyku C program fungující jako hodiny s kalendářem z cyklickým přepínáním aktuálního času a data.

Výsledné řešení musí obsahovat následující funkcionalitu:

- Softwarovou implementaci SPI komunikace s řadičem displejů MAX7219 s vhodným aplikačním rozhraním
- Softwarovou implementaci aktuálního času a data běžící od času nastaveného v kódu aplikace v proměnných.
- Každých 10s přepnutí mezi zobrazením aktuálního času s rozlišením sekund a kalendářním datem (celé datum vč. roku a dne v týdnu) doprovázené zvoleným efektem (např. odrolování / posunutí / prolnutí).
- Podporu zobrazení potřebných znaků a čísel s vhodnou reprezentací na úrovni kódu. Celý font nemusíte počítat ručně, využijte libovolný z dostupných (nebo svých) nástrojů. Musíte však vědět jak formát znaku vypadá pro případnou ruční modifikaci a nepřidělat si práci zbytečnou dodatečnou konverzí připravených dat (např. horizontálním a vertikálním zrcadlením) při použití generovaných dat.

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

8. Automatizované zpracování dat – pokladní systém

S využitím periferie čtečky čárového kódu sestavte C# aplikaci, která ve spojení s SQLite databází bude fungovat jako pokladní systém pro malou prodejnu. Důraz kladte na rychlost a jednoduchost obsluhy tohoto systému spolu s podporou pro inventarizaci zboží dostupného k prodeji. Implementujte níže požadovanou funkcionalitu, editor skladových dat nemusí být součástí řešení.

- Jednoduché a časově efektivní načtení zákazníkem vybraného zboží. Uvažujte zjednodušený terminál se čtečkou a *numerickým blokem klávesnice* bez možnosti manuálního vyhledávání.
- Korektní okamžitou reakci na neznámou nebo nedostupnou skladovou položku.
- Po ukončení prodeje se zobrazí celková cena nákupu, pokladní převezme hotovost a vrátí odpovídající částku spolu s účtenkou se všemi odpovídajícími náležitostmi. Tisk účtenky simulujte vytvořením lokálního textového souboru.
- Veškerá data a jejich změny jsou současně ukládány do databáze. Uvažujte např. kontrolu předložené účtenky při řešení reklamace, současný běh několika pokladních míst nebo vývoj ceny zboží v čase.
- Hodnoceno je mj. uživatelské prostředí a použitelnost aplikace, odolnost proti chybám a dodržení alespoň základních principů pro korektní návrh struktury tabulek databáze.