

Závěrečný projekt REV:

- Závěrečný projekt do předmětu REV-Aplikace Embedded Systémů v Mechatronice se generuje automaticky pomocí REVprojekt.exe.
- Vstupem programu je pouze číslo vašeho EDUKITU. Vygeneruje se popis stavového automatu.
- Každý stav má svoji funkcionalitu a přechody.
- Aktuální stav vždy vypisujete na první řádek displeje. Toto neplatí pro hry využívající displej.
- Mezi jednotlivými stavy se bude přecházet pomocí menu ovládaného tlačítky
 - o BTN1 – Posun kurzoru nahoru
 - o BTN2 – Posun kurzoru dolů
 - o BTN3 – Výběr stavu
 - o BTN4 – Ukončení stavu (Neplatí pro všechny hry)
- V menu zobrazte kurzor, který ukazuje na aktuální volbu
- Ukázka výsledného závěrečného projektu je k dispozici v podobě videa dostupného z GitHubu
- Posílejte pouze funkční kód, který lze zkompileovat a nahrát do mikrokontroléru

Úlohy:

GPIO:

- **Knight rider** – Na diodách zobrazujte následovný efekt, na příkladu: 000 → 001 → 011 → 111 → 011 → 001
- **Had** – je složen ze dvou diod plazí se tam a zase zpátky.
- **Tetris** – dioda odpovídá jedné kostce, které postupně připadávají, než narazí na konec, nebo předešlou kostku. Kostka se pohybuje. Tedy první projde všemi led a skončí na konci. Vše se opakuje.
- **Blikání s různou periodou** – Blikají všechny LED a to s viditelně různou periodou. Realizujte alespoň 3 různé frekvence blikání, mezi nimi přepínáte pomocí BTN2.
- **Binární čítač** – ledky reprezentují binární čítač inkrementuje BTN2
- **Postupné rozsvěcování ledek** – LED 1-6 se rozsvěčují. BTN2 přidá další led. Na konci od znovu.
- **Blikání SOS** – všechny ledky blikají viditelně SOS dle morseovky

UART:

- **Odesílání řetězce naopak** – řetězec z PC, ukončený znakem „new line“ pošlete zpět obráceně
- **Echo vrácení znaku** – vrátí do PC každý znak, který pošlete z terminálu, ten se objeví i na druhém řádku displeji
- **Výpis zprávy na displej** – odeslaný řetězec ukončený „new line“ vypíše na druhý řádek displeje
- **Volba stavu přes UART** – Zprávou z PC můžete přejít do libovolného stavu 0 až 6
- **Převod znaků na morseovku (ABCDE)** - jen znaky v závorce. Znak pošlete zpět do PC. Použijte znaky „.“ a „-“
- **Mód kalkulačka sečtení/odečtení dvou čísel** – stačí čísla typu uint8_t 0 až 99

ADC:

- **Výpis pot1 a pot2 na displej ve V** – Vypište na druhý řádek displeje napětí ve voltech ve formátu „POT1:X.XPOT2:Y.Y“

- **Ovládaní bargraph (LED) POT1** – myslí se tím sloupcový graf viz cviko ADC
- **Ovládání bargraph** – vyplňuje jeden řádek displeje vhodným znakem vytvářejte sloupcový graf na druhém řádku displeje
- **Přepnutí svítí nesvítí v polovině rozsahu potenciometru** – přepínejte stav LED 3
- **Kombinace obou kanálů ADC POT1 a POT2** – desetibitový výsledek z jednoho kanálu zkombinujte s výsledkem z druhého kanálu. Výsledek je tedy uint20 ve formátu POT2POT1. Výslednou hodnotu pošlete přes UART do PC
- **Pot1 v rozlišení 8 bit násobí hodnotu POT2 10 bit** – maximum je tedy 255x1023. Výslednou hodnotu pošlete přes UART do PC
- **Teploměr** – zobrazte na displeji teplotu z čidla na REVkitu, toto čidlo je připojeno na pin RA1/AN1. Čidlo je typu MCP9700AT-E/TT. Teplotu zobrazte ve stupních °C. Přepoččet si dohledejte v datasheetu

DAC:

- **Signálový generátor** – Generujte tři typy signálů (trojúhelník, sinusovka, klesající pila) na pinu ANALOG OUT 2, mezi jednotlivými signály přepínejte pomocí tlačítek BTN1, BTN2 a BTN3. Výstup napojte na pin RB0, zde signál digitalizujte a pošlete přes UART do počítače, ve formátu podporovaném Serial oscilloscopem. Zesílení DAC nastavte na 1x.
- **Generátor trojúhelníkového signálu** – Generujte trojúhelníkový signál s proměnlivou frekvencí a amplitudou. Pomocí POT1 měňte frekvenci signálu a pomocí POT2 jeho amplitudu. Rozsahy frekvence a amplitudy si vhodně zvolte sami. Signále generujte na pin ANALOG OUT2. Výstup napojte na pin RB5, zde signál digitalizujte a pošlete přes UART do počítače, ve formátu podporovaném Serial oscilloscopem. Zesílení DAC nastavte na 1x.
- **Generátor půlkruhového signálu** – Generujte signál, který bude složen z půlkružnic na způsob sinusovky (www). Parametry signálu si vhodně zvolte sami. Signále generujte na pin ANALOG OUT2. Výstup napojte na pin RB5, zde signál digitalizujte a pošlete přes UART do počítače, ve formátu podporovaném Serial oscilloscopem. Zesílení DAC nastavte na 1x.
- **Simulace spínání triaku** – Generujte sinusový signál a pomocí potenciometru měňte dobu zapnutí virtuálního triaku. Výsledný signál bude vypadat následovně (www), moment zapnutí triaku bude nastavitelný pomocí POT2. Frekvence sinusovky bude 50 Hz. Signále generujte na pin ANALOG OUT1. Výstup napojte na pin RB0, zde signál digitalizujte a pošlete přes UART do počítače, ve formátu podporovaném Serial oscilloscopem. Zesílení DAC nastavte na 1x.
- **Ořezaná sinusovka** – Generujte sinusový signál a pomocí POT1 a POT2 nastavujte horní a dolní saturaci sinusovky. POT2 bude signál ořezávat od maximální hodnoty sinusovky do střední hodnoty a POT1 od nejvyšší hodnoty sinusovky po střední hodnotu. Využijte celý rozsah potenciometrů. Signále generujte na pin ANALOG OUT1. Výstup napojte na pin RB0, zde signál digitalizujte a pošlete přes UART do počítače, ve formátu podporovaném Serial oscilloscopem. Zesílení DAC nastavte na 1x.

PWM:

- **Ovládaní rychlosti motoru a LED5 potenciometrem POT1 LED5 je stejná** – PWM na LED je stejná jako na motoru (když LED svítí LED na max, tak i motor se otáčí maximálními otáčkami)
- **Ovládaní rychlosti motoru a LED5 potenciometrem POT1 LED5 je opačná** – PWM na LED je opačná než na motoru (když LED svítí LED na max, tak motor stojí)
- **Ovládaní rychlosti motoru potenciometrem bez LED** – Na LED není PWM
- **PWM + BTN2** – BTN2 přepíná jas LED mezi pěti úrovněmi. Myslí se tím 5 úrovní jasu LED
- **PWM-blikání s plynulou změnou jasu (sinus)** - jas led je funkcí $\sin(2\pi \cdot f)$ Změna je viditelná. Jas se mění sinusově od nuly po max hodnotu.

- **PWM-blikání s plynulou změnou jasu (trojúhelník)** – Jas LED se mění lineárně od nuly po max hodnotu.
- **Ovládání rychlosti motoru tlačítky** – BTN 1 start/stop, BTN 2 směr otáčení (zachovejte stejnou rychlost otáčení při změně směru), BTN 3 ubrat střidu, BTN 4 přidat střidu

HRY:

- **Závody BTN:**
 - Pomocí tlačítka BTN1 a BTN2 přepínáte polohu vozidla mezi řádky
 - Vozidlo zobrazujete jako znak „>“, zůstává v prvním sloupci
 - Překážky se pohybují směrem k autu
 - Překážky generujete náhodně pomocí funkce rand()
 - Hráč má tři životy, jejich počet zobrazte pomocí LED
 - Při zásahu překážky dojde k ubrání životu
 - Po třech zásazích překážky zobrazte na 2 vteřiny „GAME OVER“ na prvním řádku a skóre na řádku druhém (ujetá vzdálenost vyjádřená v počtu znaků displeje)
 - Následně přejděte zpět do menu
- **Závody POT:**
 - Pomocí POT 2 přepínáte polohu vozidla mezi řádky
 - Vozidlo zobrazujete jako znak „>“, zůstává v prvním sloupci
 - Překážky se pohybují směrem k autu
 - Překážky generujete náhodně pomocí funkce rand()
 - Hráč má tři životy, jejich počet zobrazte pomocí LED
 - Při zásahu překážky dojde k ubrání životu
 - Po třech zásazích překážky zobrazte na 2 vteřiny „GAME OVER“ na prvním řádku a skóre na řádku druhém (ujetá vzdálenost vyjádřená v počtu znaků displeje)
 - Následně přejděte zpět do menu
- **1D Pong:**
 - Jde o hru pro dva hráče. Hráč 1 ovládá BTN1, hráč 2 ovládá BTN4
 - Po displeji se bude na 2. řádku pohybovat znak reprezentující míček „o“.
 - Na koncích 2. řádku budou zobrazeny dva znaky ve formě bloků „■“ (ASCII 219) reprezentující pátky pro odpálení míčku.
 - Hráč musí zmáčknout tlačítko v době, kdy se míček nachází před pálkou, aby došlo k jeho odpálení na druhou stranu.
 - Pokud zmáčkne tlačítko dříve, tak se nic neděje.
 - Pokud nestihne zmáčknout tlačítko v požadované pozici, tak hráč ztrácí život.
 - Každý hráč má tři životy, které jsou zobrazovány na prvním řádku displeje.
 - Pokud jeden hráč dosáhne na nula životů zobrazí se na 2 vteřiny „GAME OVER“ na prvním řádku a „Player 1/2 won“ na řádku druhém
 - Následně se opět přejde do menu
- **Guitar Hero:**
 - Po displeji budou zprava do leva jezdit náhodně generované číslice 1 až 4 odpovídající tlačítkům na REVkitu, využijte funkci rand()
 - Hráč musí zmáčknout správná tlačítka ve chvíli, kdy se nachází na prvním segmentu na levé straně displeje.
 - Pokud zmáčkne jiná tlačítka, než má nebo tlačítka nestihne zmáčknout, tak přijde o jeden život, za každé špatné zmáčknutí nebo nezmáčknutí.
 - Celkem má hráč tři životy, které jsou signalizovány pomocí LED

- Při dosažení nula životů se na prvním řádku zobrazí „GAME OVER“ a na druhém skóre, které odpovídá počtu správně stisknutých tlačítek.
- Následně se přejde zpět do menu
- **Ladění potáku:**
 - Na prvním řádku je zobrazeno náhodné číslo v rozsahu 0 až 1023. Na druhém řádku je zobrazena aktuální hodnota získaná z POT 1.
 - Cílem je v daném časovém intervalu nastavit stejnou hodnotu na POT1, jaká je zobrazena na displeji v prvním řádku.
 - Časový limit pro nastavení potenciometru je odpočítávám pomocí LED.
 - Po každém kole se časový limit zkracuje.
 - V případě, že vyprší čas, než je nastavena správná poloha, tak dojde k ukončení hry. Na prvním řádku zobrazí „GAME OVER“ a na druhém skóre, po dvou vteřinách se opět vrátí hráč do menu.
- **Uhodni číslo:**
 - Náhodně vygenerujte číslo od 0 do 100 pomocí funkce rand()
 - Hráč musí uhodnout dané číslo zadáním svého odhadu do Termitu.
 - PIC odpovídá pouze zprávou, zda je zadané číslo vyšší nebo nižší než číslo, které má být uhodnuto.
 - Hráč má na uhádnutí čísla omezený počet pokusů. Zbývající pokusy jsou signalizovány pomocí LED
 - Na displeji bude zobrazen název hry.
 - Při výhře nebo prohře se hráč vrátí zpět do menu.
 - Ke hře realizujte vhodné textové uživatelské rozhraní, které se vypíše do konzole. Hra by měla být intuitivní a dávat vhodné instrukce.
- **Rychlost reakce:**
 - Na displeji se na prvním řádku zobrazí název hry.
 - Na druhém řádku se v náhodný okamžik zobrazí číslo 1 až 4, hráč musí, co nejrychleji stisknout tlačítko odpovídající zobrazenému číslu. Využijte funkci rand().
 - Po stisknutí tlačítka se na displeji zobrazí reakční doba v ms (zobrazte také jednotky). Po dvou vteřinách se vraťte zpět do menu.
 - Ošetřete stavy, kdy nedojde ke zmáčknutí žádného tlačítka.
- **Odhad času:**
 - Cílem hry je co nejpresněji odhadnout zadaný časový okamžik.
 - Na displeji se zobrazí na prvním řádku doba v sekundách, kterou má člověk odhadnout (3 až 20 s). Tato doba se generuje náhodně pomocí funkce rand()
 - Pro začátek hry musí hráč stisknout tlačítko 1. Tímto tlačítkem také ukončí hru, a následně se na displeji zobrazí chyba odhadu v ms.
 - Po dvou vteřinách následuje návrat do menu.
- **LOVE machine:**
 - Automat bude mít textové rozhraní realizované přes UART, které bude uživatele informovat, co má dělat.
 - Nejdříve je nutné vložit přes UART minci ve formátu ***coin***
 - Následně bude uživatel vyzván k přiložení prstu na teploměr.
 - Po uplynutí určité doby se vypíše míra zamilovanosti přes UART spolu se změřenou teplotou.
 - Vytvořte alespoň 5 stavů zamilovanosti a rozvrhněte je rovnoměrně v teplotním intervalu, který experimentálně určíte. Kreativně se meze nekladou.

- Po vypsání stavu zamilovanosti a teploty dojde po pěti vteřinách k návratu do menu.