

# Závěrečný projekt REV:

- Závěrečný projekt do předmětu REV – Aplikace Embedded Systémů v Mechatronice se generuje automaticky pomocí REVprojekt.exe nebo pomocí online interpretu.
  - o Vstupem programu je vaše studentské číslo.
  - o Vygenerují se jednotlivé úkoly vašeho projektu.
- Projekt každý student vypracovává **samostatně**. Není možné používat již hotové cizí kódy s výjimkou příkladů na githubu a lcd knihovny.
- Každý stav má svoji funkcionalitu a přechody.
- Aktuální stav vždy vypisujte na první řádek displeje. Toto neplatí pro hry využívající displej nebo pokud není dáno jinak.
- Mezi jednotlivými stavy se bude přecházet pomocí menu ovládaného tlačítka
  - 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
- Při pohybu v menu implementujte scrollování jednotlivých položek, jedna z možných interpretací je na videu níže.
- Pro správnou odezvu na stisk tlačítek využijte nějakou metodu debouncingu.
- Výsledný program by měl vždy téměř okamžitě reagovat na stisk tlačítka. Snažte se tedy vyvarovat používání příliš dlouhých delay funkcí.
- Motor se má točit pouze v úlohách, ve kterých je to požadováno.
- Ukázka výsledného závěrečného projektu je k dispozici v podobě videa dostupného z [GitHubu](#)

## Odevzdání:

- Odevzdávejte pouze funkční projekt vytvořený v programu MPLAB, který lze zkompileovat a nahrát do mikrokontroléru.
- Obvod vytvořený v HW sekci vyfotografujte společně s REVkitem a fotografii odevzdejte s projektem. Pokud vytvořený obvod nevychází z obvodů dostupných na GitHubu předmětu, tak vytvořte i jeho schéma obsahující hodnoty jednotlivých komponent (ručně nebo na PC) a to přiložte k odevzdanému projektu.
- Výslednou složku s projektem a fotografií umístěte do složky *Prijmeni\_Jmeno\_VUTID* a tu zazipujte. Zip pojmenujte stejně, jako složku.

## Hodnocení:

- Maximální možný počet bodů je 50.
- Hodnocení dílčích částí:
  - o Menu: 8 b
  - o GPIO: 4 b
  - o UART: 4 b
  - o ADC: 4 b
  - o DAC: 6 b
  - o PWM: 4 b
  - o HRY: 10 b

- HW: 10 b
- Menu je nutnou součástí projektu. Projekt nelze odevzdat jako samostatné úlohy.
- Detaily hodnocení:
  - Projekt, který nelze zkompileovat a nahrát do mikrokontroléru je hodnocen nula body.
  - Pokud je uživatelské rozhraní špatně ovladatelné (reaguje pomalu, nedostatečný debouncing, příliš zjednodušené menu) jsou body odečítány z hodnocení menu.
  - Pokud nějaká úloha nevykazuje žádnou známku funkčnosti, je hodnocena nula body. Na podobu zdrojového kódu není brán zřetel.
  - V případě alespoň částečné funkčnosti úlohy je udělen odpovídající počet bodů.
  - Kvalita zdrojového kódu není hodnocena.

## Úlohy:

### GPIO:

- **Knight rider** – Na všech šesti LED zobrazujte následovný efekt, na příkladu pro tři LED: 000 → 001 → 011 → 111 → 011 → 001
- **Had** – je složen ze LED dvou diod a plazí se nakonec všech diod a zase zpět. V každém kroku se posune vždy o jednu diodu. Př: 1100 → 0110 → 0011 → 0110 → 1100 ...
- **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 (1 Hz, 4 Hz a 10 Hz), mezi nimi přepínejte pomocí BTN2.
- **Binární čítač** – ledky reprezentují binární čítač inkrementuje pomocí BTN2. Využijte všech šest LED.
- **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:

Baudrate u všech úloh nastavte na [19 200](#).

- **Odesílání řetězce naopak** – řetězec z PC, ukončený znakem „new line“ pošlete zpět obráceně. Maximální možná délka řetězce je 30 znaků.
- **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 znakem „new line“ vypíše na druhý řádek displeje.
- **Volba stavu přes UART** – Zprávou z PC můžete přejít do libovolného stavu 1 až 6. Zpráva bude ve formátu „FX“, kdy X je číslo stavu od 1 do 6 a zpráva je ukončena znakem „new line“.
- **Převod znaků na morseovku** – Převádějte vstupní znaky na jejich odpovídající znaky v Morseově abecedě. Vstupní znaky uvažujte pouze A, B, C, D a E. Znak v morseově abecedě pošlete zpět do PC. Použijte znaky „.“ a „-“. Vstupní znaky zadávané do terminálu budou ukončeny znakem „new line“. Znaky jsou zadávány pouze po jednom.
- **Mód kalkulačka sečtení/odečtení dvou čísel** – vstup bude v podobě výrazu A+B= nebo A-B= následovaný znakem „new line“. Vstupní čísla budou pouze v rozsahu 0 až 99. Výsledek může tedy být i záporný.

#### 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** – Podle hodnoty z POT1 postupně v pravidelných intervalech rozsvěcujte všechny LED.
- **Ovládaní bargraph (LCD) POT1** – Podle hodnoty z POT1 postupně v pravidelných intervalech vyplňte druhý řádek displeje vámi zvoleným znakem.
- **Přepnutí svítí nesvítí v polovině rozsahu potenciometru** – přepínejte stav LED 3 podle hodnoty POT 1 a LED 5 podle hodnoty POT 2. V dolní polovině rozsahu LED nesvítí a naopak.
- **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. Hodnotu posílejte opakovaně.
- **Pot1 v rozlišení 8 bit násobí hodnotu POT2 10 bit** – maximum je tedy 255x1023. Výslednou hodnotu pošlete přes UART do PC. Hodnotu posílejte opakovaně.
- **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 s přesností na desetiny, např 26,2 °C. Přepočít si dohledujte v datasheetu teploměru.

#### 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 [RBO](#), 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 (signál jde nahoru a dolů, nejedná se o pilu!) s proměnlivou frekvencí a amplitudou. Pomocí POT1 měňte frekvenci signálu a pomocí POT2 jeho amplitudu. Rozsahy frekvence a amplitudy si vhodně zvolte sami. Signály generujte na pin [ANALOG OUT2](#). Výstup napojte na pin [RBO](#), 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. Signály generujte na pin [ANALOG OUT2](#). Výstup napojte na pin [RBO](#), 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. Signál generujte na pinu [ANALOG OUT1](#). Výstup napojte na pin [RB5](#), 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 nejnižší hodnoty sinusovky po střední hodnotu. Využijte celý rozsah potenciometrů. Signál generujte na pin [ANALOG OUT1](#). Výstup napojte na pin [RB5](#), 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 na max, tak i motor se otáčí maximálními otáčkami). Po ukončení stavu se motor zastaví.
- **Ovládaní rychlosti motoru a LED5 potenciometrem POT1, LED5 je opačná** – PWM na LED je opačná než na motoru (když LED svítí na max, tak motor stojí). Po ukončení stavu se motor zastaví.
- **Ovládaní rychlosti motoru potenciometrem POT 2 bez LED** – V polovině rozsahu motor stojí. Poté se podle hodnoty na POT2 roztáčí na jednu stranu a na druhou. Po ukončení stavu se motor zastaví.
- **PWM + BTN2** – BTN2 přepíná jas LED mezi pěti úrovněmi. Myslí se tím 5 úrovní jasu LED.
- **PWM-blikání LED5 s plynulou změnou jasu (trojúhelník)** – Jas LED se mění lineárně od nuly po max hodnotu a zpět do nuly.
- **Ovládaní rychlosti motoru tlačítky** – BTN 1 start/stop, BTN 2 směr otáčení (zachovejte stejnou rychlost otáčení při změně směru), POT2 střída. Po ukončení stavu se motor zastaví.

## HRY:

- **Závody BTN:**
  - o Pomocí tlačítka BTN1 a BTN2 přepínáte polohu vozidla mezi řádky LCD
  - o Vozidlo zobrazujete jako znak „>“, zůstává v prvním sloupci
  - o Překážky se pohybují směrem k autu
  - o Překážky generujete náhodně pomocí funkce rand()
  - o Hráč má tři životy, jejich počet zobrazte pomocí LED
  - o Při zásahu překážky dojde k ubrání životu
  - o 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)
  - o Následně přejděte zpět do menu
- **Závody POT:**
  - o Pomocí POT 2 přepínáte polohu vozidla mezi řádky
  - o Vozidlo zobrazujete jako znak „>“, zůstává v prvním sloupci
  - o Překážky se pohybují směrem k autu
  - o Překážky generujete náhodně pomocí funkce rand()
  - o Hráč má tři životy, jejich počet zobrazte pomocí LED
  - o Při zásahu překážky dojde k ubrání životu
  - o 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)
  - o Následně přejděte zpět do menu
- **1D Pong:**
  - o Jde o hru pro dva hráče. Hráč 1 ovládá BTN1, hráč 2 ovládá BTN4
  - o Po displeji se bude na 2. řádku pohybovat znak reprezentující míček „o“.
  - o Na koncích 2. řádku budou zobrazeny dva znaky ve formě bloků „■“ (ASCII 219) reprezentující pátky pro odpálení míčku.
  - o Hráč musí zmáčknout tlačítko v době, kdy se míček nachází před pálkou, aby došlo k jeho odpálení na druhou stranu.
  - o Pokud zmáčkne tlačítko dříve, tak se nic neděje.
  - o Pokud nestihne zmáčknout tlačítko v požadované pozici, tak hráč ztrácí život.
  - o 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. Posílá se číslo a znak „new line“.
  - 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 nejpřesněji odhadnout zadaný časový okamžik.
  - Na displeji se zobrazí na prvním řádku doba v sekundách, kterou má člověk odhadnout (3 až 10 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\*** plus znak „new line“.
  - 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.

## HW úkol:

- **Přehrávač hudby – timer:**
  - Vytvořte hudební přehrávač, který bude přehrávat polyfonní skladbu na připojeném reproduktoru.
  - Tlačítkem **BTN1** zastavte přehrávání, dalším stiskem tlačítka přehrávání obnovte ze stejného bodu.
  - Signál pro spínání tranzistoru generujte na základě timeru.
  - Reprodukter připojte na pin **RB5**.
  - Zapojení a skladbu použijte z cvičení BUT-FME-REV/0X\_HW\_cv/02\_Timer\_ISR/
- **Přehrávač hudby – pwm:**
  - Vytvořte hudební přehrávač, který bude přehrávat polyfonní skladbu na připojeném reproduktoru pomocí změny frekvence PWM signálu.
  - Tlačítkem **BTN2** změňte přehrávanou skladbu přepínejte mezi dvěma skladbami.
  - Signál pro spínání tranzistoru generujte pomocí PWM periferie na pinu **RB5**.
  - PWM periferie neumožňuje přehrát noty s velmi nízkou frekvencí, proto frekvenci všech not vynásobte **dvakrát** a vhodně nastavte Timer používaný pro generování frekvence PWM.
  - Zapojení a skladbu použijte z cvičení BUT-FME-REV/0X\_HW\_cv/02\_Timer\_ISR/. Další skladbu si zvolte dle vlastní preference na stránce na odkazu dostupném v souboru notes.h
- **Dálkoměr:**
  - S využitím IR diody a IR tranzistoru sestrojte jednoduchý dálkoměr. Využijte obvod dostupný na GitHubu předmětu REV (BUT-FME-REV/0X\_HW\_cv/04\_ADC/).
  - Analogové napětí měřte pomocí pinu **RB5**.
  - Dálkoměr kalibrujte v rozsahu cca 5 – 15 cm pomocí bílého kancelářského papíru a výslednou vzdálenost vypisujte na displej v cm.
- **Neinvertující sčítač:**
  - S využitím operačního zesilovače sestrojte obvod neinvertujícího sčítače. Vstupem do obvodu budou dva kanály DA převodníku a výstup bude veden na pin **RB5**.
  - Výstupní napětí na jednotlivých kanálech DA převodníku bude nastavováno potenciometry 1 a 2 (POT2 -> Analog Out1, POT1 -> Analog Out2). Výstupní zesílení DA převodníku nastavte na 1x.

- Parametry neinvertujícího sčítače nastavte tak, aby výstupní napětí bylo součtem vstupních napětí.
- Na displej vypište jednotlivá napětí generovaná DA převodníkem v mV a hodnotu napětí měřenou na pinu **RB5** také v mV.
- **Rozdílový operační zesilovač:**
  - S využitím operačního zesilovače sestrojte rozdílové zapojení. Vstupem do obvodu budou dva kanály DA převodníku a výstup bude veden na pin **RB5**.
  - Výstupní napětí na jednotlivých kanálech DA převodníku bude nastavováno potenciometry 1 a 2 (POT2 -> Analog Out1, POT1 -> Analog Out2). Výstupní zesílení DA převodníku nastavte na 1x.
  - Parametry zapojení nastavte tak, aby výstupní napětí bylo rozdílem napětí z Analog Out1 a Analog Out2 (AO1 – AO2).
  - Na displej vypište jednotlivá napětí generovaná DA převodníkem v mV a hodnotu napětí měřenou na pinu **RB5** také v mV.
- **DIY optočlen:**
  - Vytvořte si vlastní optočlen pomocí IR LED a IR tranzistoru. IR tranzistorem poté spínejte připojenou externí LED.
  - Vstupní signál do optočlenu bude z tlačítka BTN 1 a bude připojen na pin **RB5**.