

Dokumentace k projektu z předmětu NAV

Vestavěný systém pro měření vzdálenosti (ultrazvukový senzor)

Michal Rein

10. dubna 2022

1 Platforma

Projekt je koncipován tak, aby fungoval na výukovém kitu *FITkit v3.0* s mikrokontrolérem Kinetis-K60 od společnosti Freescale Semiconductor s jádrem ARM Cortex-M4. Pro vývoj na této platformě lze využít vývojové prostředí *Kinetis Design Studio*, ve kterém byl tento projekt rovněž vyvíjen. Na kitu se nachází header pinů s označením P1, na který jsou namapovány jednotlivé vývody z MCU. K připojení komponent je tedy využít právě header P1 podle schématu na obrázku 4.

2 Komponenty a jejich zapojení

2.1 LCD displej LFD039AUE-102A

Pro zobrazování výsledků měření slouží jednoduchý LCD displej LFD039AUE-102, který dokáže zobrazovat až čtyřmístná čísla i s desetinnou čárkou. Schéma displeje je možné vidět na obrázku 1, jednotlivé propojení vstupů na segmenty a sloupce je na obrázku 2. Vstupy 12, 9, 8, 6 pak slouží k selekci sloupce a 11, 7, 4, 2, 1, 10, 5, 3 k selekci jednotlivých segmentů. Všechny tyto vstupy jsou vyvedeny do jednotného konektoru s označením JP1 dle schématu 3. Tento konektor je konečně připojen na header pinů P1, přičemž selekci sloupců displeje je možné provádět piny s označením [17–20], Konkrétní segmenty jsou pak připojeny k pinům [21–28]. Na schématu 4 lze pak vidět konkrétní umístění konektoru JP1, včetně popisu jednotlivých vstupů konektoru mapovaných podle schématu 3 a popisu portů připojených k MCU.

2.2 Ultrazvukový měřič vzdálenosti HY-SRF05

K měření vzdálenosti je využit senzor HY-SRF05. Senzor má celkově 4 vývody, přičemž 2 je nutné připojit na zdroj napětí 5V a zem, zbylé 2 nesou označení **Trig** a **Echo**. **Trig** slouží pro zahájení povelu měření ze strany MCU, **Echo** pak ke snímání naměřené hodnoty. Výstup však není v digitální podobě, nýbrž je nutné měřit dobu, po kterou je výstup **Echo** v hodnotě logické 1 a následně tuto dobu převést na výslednou hodnotu dle vztahu $d = (d_{usec} * t) / 2$, kde d_{usec} je vzdálenost, kterou zvuk urazí za mikrosekundu, vyjádřená vztahem $d_{usec} = c / 10000$, přičemž c je rychlost šíření zvuku (přibližně 343m/s) a t doba v mikrosekundách, po kterou měření trvalo.

Pro zahájení měření je potřeba přivést na vstup **Trig** puls o velikosti alespoň 10us. Zařízení následně provede měření a vystaví na výstup **Echo** signál, jehož délku je nutné změřit. Podle specifikace¹ modelu HY-SRF05 je zařízení schopno měřit vzdálenost v rozmezí 2–450cm a odchylkou přibližně 3mm. Mapování vývodů modulu na header P1 lze nalézt na obrázku 4 pod značkami +5V, GND, **Trigger** a **Echo**.

3 Uživatelské rozhraní

Navržený vestavěný systém mimo samotné měření vzdálenosti a zobrazení výsledků na LCD displej podporuje rovněž možnost volby jednotek, ve kterých je výsledek zobrazován.

¹<http://www.datasheet-pdf.com/PDF/HY-SRF05-Datasheet-ETC-813041>

Systém podporuje celkem 3 jednotky vzdálenosti, a to milimetry, centimetry a metry. Ve výchozím nastavení systém vždy zobrazuje hodnoty v centimetrech. Pro volbu jiné jednotky je možné využít tlačítko umístěné přímo na přípravku FITkit s označením SW6. Zvolený mód měření je indikován LED diodami, které jsou umístěny vedle tohoto tlačítka. Dioda s označením D12 pak indikuje měření v centimetrech, D11 v metrech a D10 v milimetrech.

4 Implementace

Projekt je napsán v programovacím jazyce C a prostředí Kinetis Design Studio. Celkově je projekt rozdělen na 2 části:

- Mapování a obsluha displeje - `display.c` + `display.h`
- Logika měření a inicializace zařízení - `main.c`

V popisu této části bude pouze nastíněna implementace systému, pro hlubší pohled je vhodné nahlédnout na samotné zdrojové soubory projektu, které jsou řádně okomentovány a slouží tak jako součást dokumentace.

4.1 Obsluha přerušení

V projektu se vyskytují 2 druhy přerušení:

1. Přerušení informující o dokončení vysílání od senzoru
2. Přerušení informující o změně jednotek hodnot zobrazených na displeji

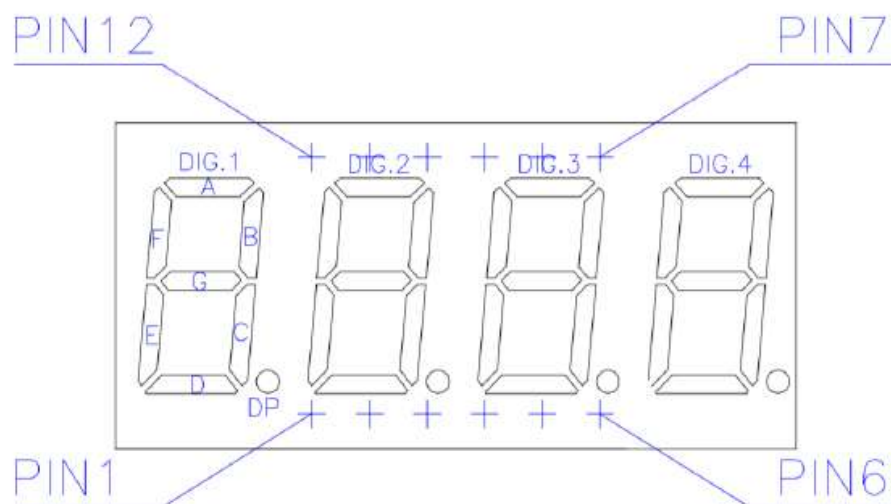
V prvním případě je zdrojem přerušení nástup sestupné hrany na pinu `Echo`, která indikuje ukončení přenosu ze strany senzoru. Doba, po kterou byl tento vstup v hodnotě logické 1, je následně k dispozici v proměnné `end`. Rovněž dochází k nastavení příznaku o vysílání `receiving_signal` na hodnotu `False`. V druhém případě se jedná o přerušení způsobené tlačítkem SW6, které umožňuje přepínání mezi konverzí jednotek naměřených hodnot.

4.2 Měření doby vysílání senzoru

Aby bylo měření co nejpresnější, je potřeba zvolit časovač, který bude schopen počítat rychlostí kolem mikrosekundy. K tomuto účelu slouží *Flex timer (FTM)*, jehož parametry a možnosti nastavení lze nalézt v dokumentaci² k mikrokontroleru Kinetis K60. Časovač je připojen na systémové hodiny. Jelikož má samotný časovač k dispozici pouze 16-bitový registr, je nutné nastavit předděličku, jinak by došlo k velmi rychlému přetečení registru a znehodnocení měření. V této implementaci je hodnota předděličky nastavena na 32. Pro kmitočet systémových hodin 50MHz je inkrementována hodnota registru každých 640ns. Registr je schopen napočítat až $2^{16} * 0.64 = 41943us$, tedy maximální měřitelná vzdálenost je přibližně $41943us * 0.0343cm/us = 1438cm$. Tato hodnota je vzhledem k maximálnímu rozsahu senzoru více než dostatečná i při skutečnosti, že je výsledná hodnota dělena 2 kvůli započítání vzdálenosti k objektu i zpět.

Měření probíhá v pravidelných intervalech a stará se o něj metoda `start_measurement(float *storage, float distance_per_usec)`. Pin připojený na vstup `Trig` je nejdříve ustálen do hodnoty logické 0. Následuje vyslání krátkého, přibližně 10us dlouhého pulzu hodnoty logické 1. Po ukončení vysílání je potřeba zastavit vykonávání do doby, než se na vstupu `Echo` objeví logická 1 od senzoru. Jakmile se tak stane, dochází k resetu počítadla FTM, nastaví se příznak příjmu signálu a dochází k iterativnímu ukládání hodnoty časovače do proměnné `end` do doby, než přijde přerušení o ukončení vysílání popsaného v podkapitole 4.1. Následuje přepočít podle vzorců popsaných v kapitole 2.2 a zobrazení výsledné hodnoty.

²https://www.nxp.com/files-static/32bit/doc/ref_manual/K60P144M100SF2RM.pdf



Obrázek 1: Schéma displeje a označení segmentů

4.3 Zobrazení dat na displej

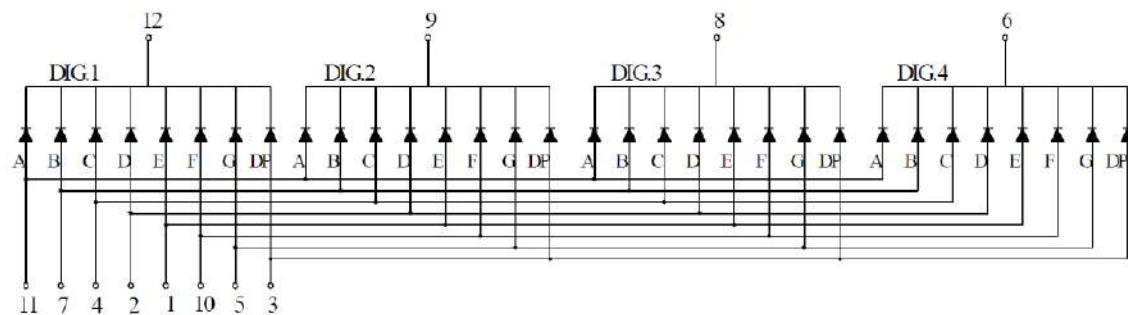
Zobrazení hodnoty na displej probíhá rovněž periodicky (avšak rychleji než měření) pomocí metody `display_measurement(float distance)`. Pro obsluhu displeje byl vytvořen modul `display`, ve kterém je nadefinováno mapování sloupců a segmentů displeje na jednotlivé piny portů a implementace samotných metod pro obsluhu, kterými jsou:

- `select_display_column(int index)` - vybere sloupec displeje
- `reset_display()` - vyčistí displej
- `show_number(int num)` - zobrazí číslo na vybraném sloupci displeje
- `show_dot()` - zobrazí tečku na vybraném sloupci displeje

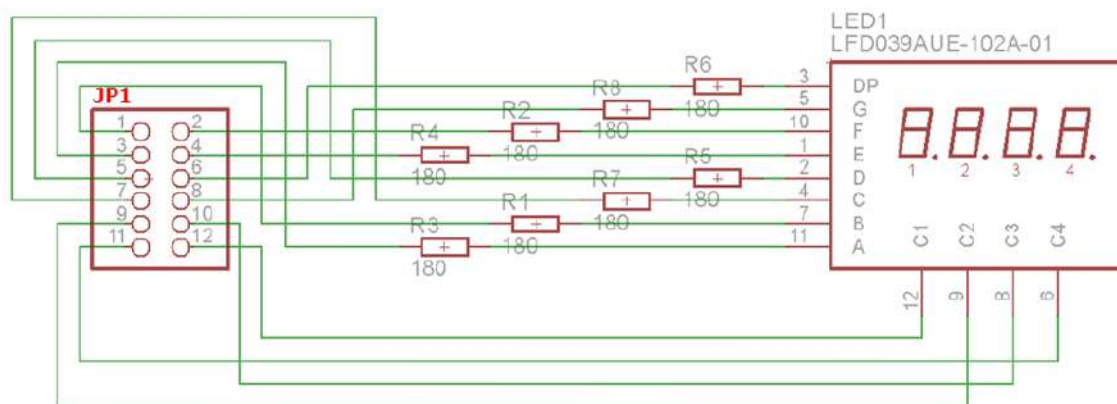
Metoda `show_number` vyžaduje číslo v celočíselném tvaru. Naměřené hodnoty datového typu `float` je tedy nutné nejdříve rozdělit na jednotlivá čísla (včetně desetinné čárky), převést na celá čísla a následně je postupně zobrazit. K rozdělení čísla slouží metoda `gcvt` z knihovny `<stdlib.h>`, která rozdělí číslo kolem desetinné čárky o požadované délce a převede jej na datový typ `char*`. Kromě znaku tečky (desetinné čárky), lze pak jednotlivé číslice typu `char` převést na odpovídající typ `int` odečtením hodnoty znaku číslice od znaku '0'.

5 Závěr

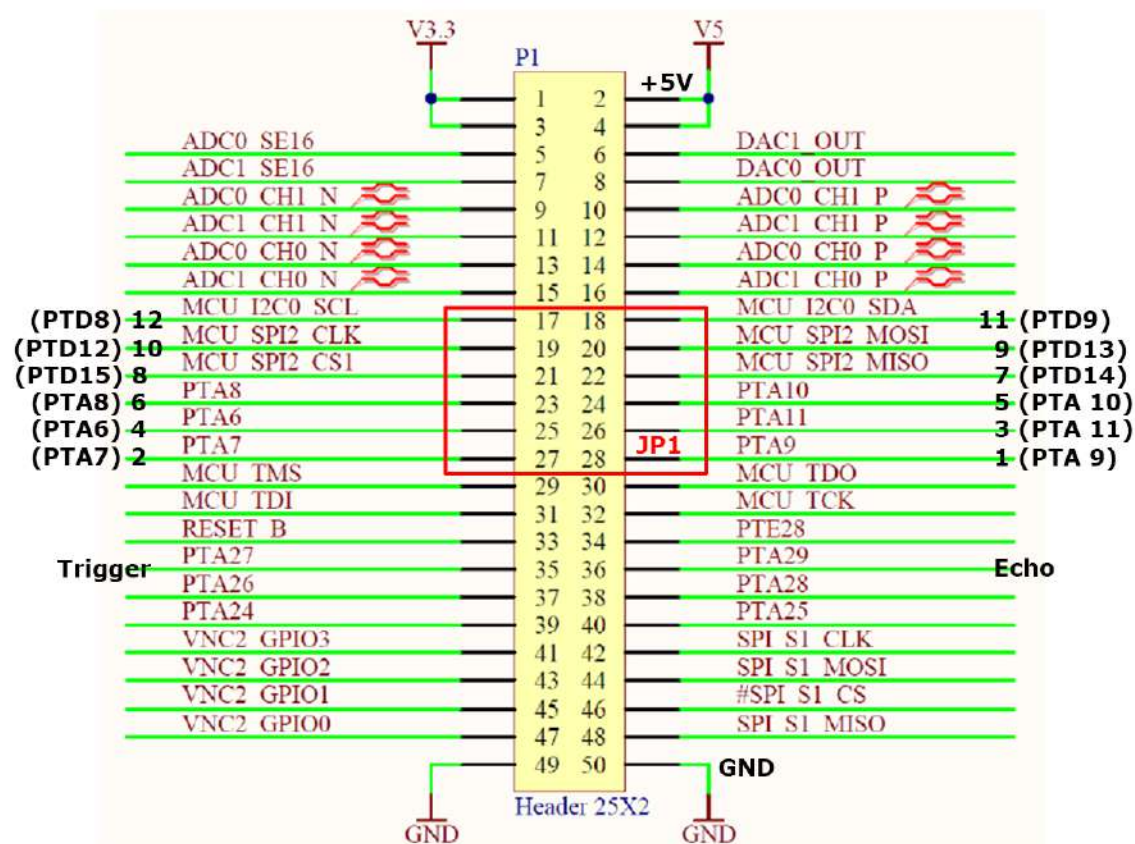
Výsledný vestavěný systém funguje dle očekávání, přesnost měření odpovídá parametrům daných výrobcem senzoru. Na obrázcích 5, 6 a 7 si lze prohlédnout příklady měření různých vzdáleností se zobrazením v různých jednotkách (cm, m, mm).



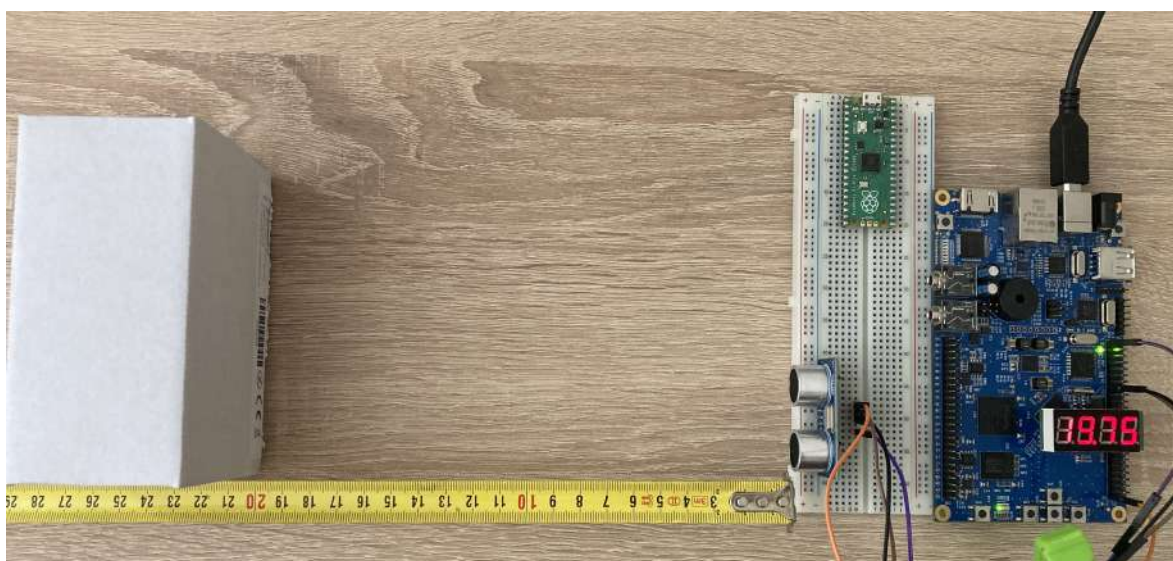
Obrázek 2: Schéma mapování vstupů na jednotlivé segmenty displeje



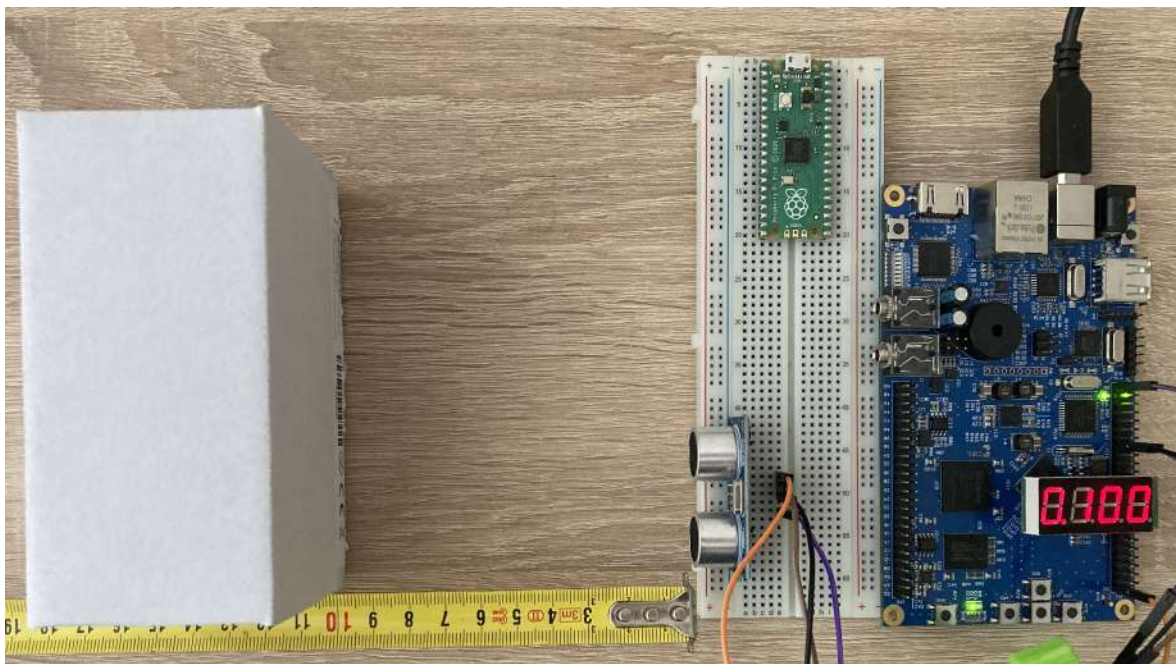
Obrázek 3: Schéma mapování částí LED displeje na výstupní konektor



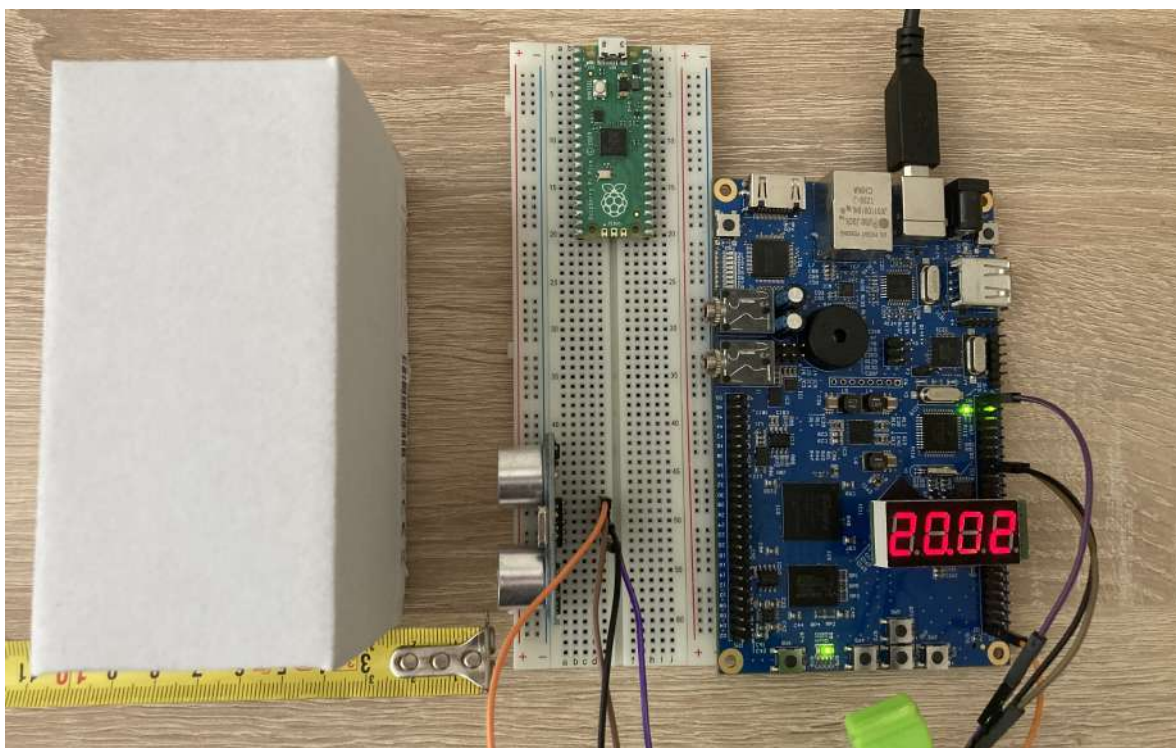
Obrázek 4: Schéma zapojení jednotlivých komponent na header P1



Obrázek 5: Příklad měření vzdálenosti 20cm v jednotkách centimetrů



Obrázek 6: Příklad měření vzdálenosti 10cm v jednotkách metrů



Obrázek 7: Příklad měření vzdálenosti 2cm v jednotkách milimetrů