

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

IMP – Projekt

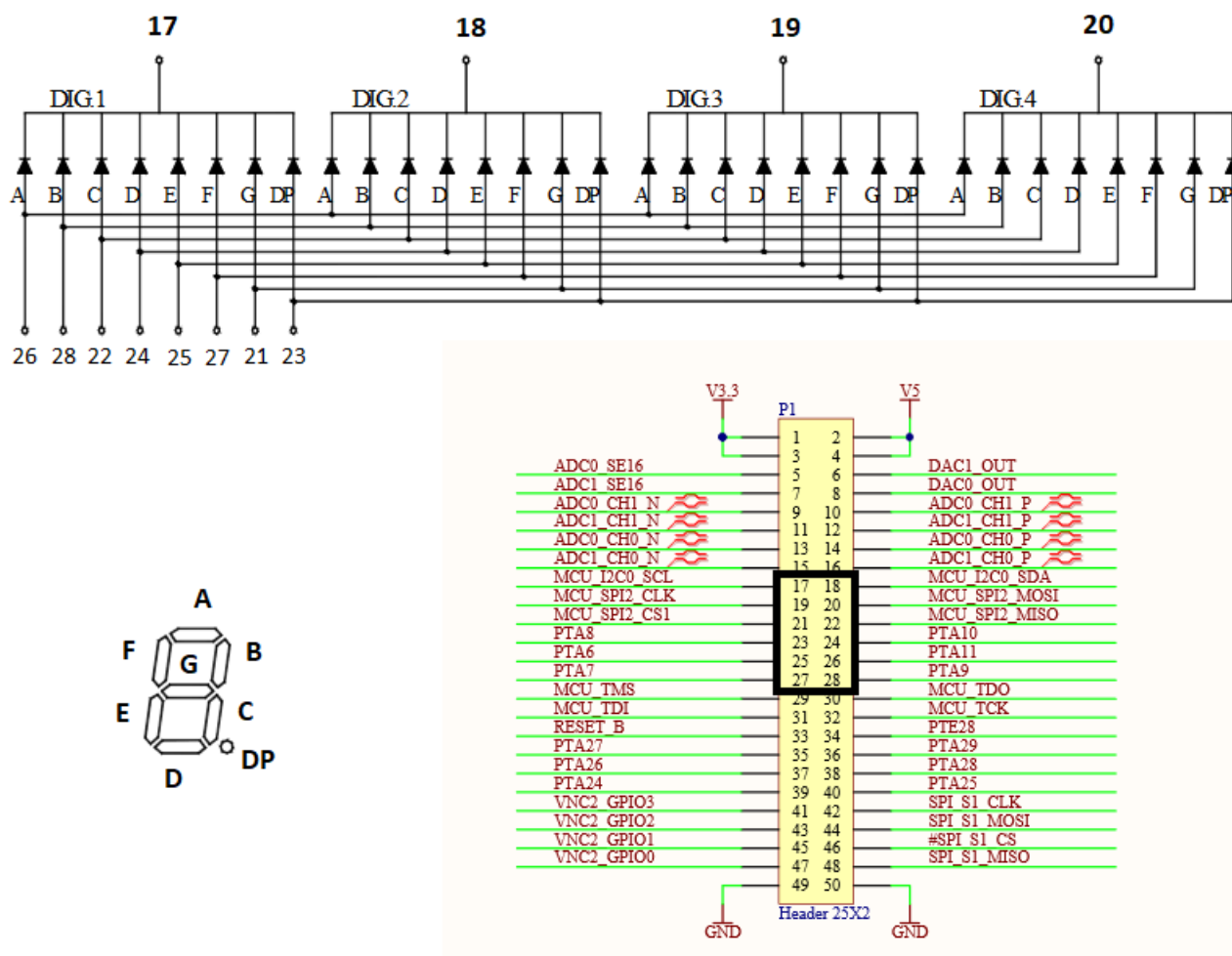
ARM-FITkit3: Měření vzdálenosti
ultrazvukovým senzorem

Úvod

Zadání měření vzdálenosti pomocí ultrazvukového měřiče jsem si vybral z důvodu, že na střední škole jsem se poměrně dost zabýval Arduinem a věcí s ním spojeným. Přímo tento měřič jsem používal k měření výšky hladiny (https://github.com/kocica/Automaticky-zavlazovací-system/blob/master/final_presentation.pdf). Projekt jsem si vybral abych pochopil a vyzkoušel si měření na nižší úrovni, než pouze zavolat abstraktní funkci z knihovny a dostat výsledek.

1 LED Display LFD039AUE-102A

LED display je zapojen v poli P1 v pinech PTA [6-11], a PTD [8-9, 12-15]. Tedy pinech na Fitkitu 17 až 28. Display je zapojen směrem "dovnitř" fitkitu viz. Obrázek 2.



Obrázek 1: Zapojení LED displaye



Obrázek 2: Ukázka zapojení LED displaye

Display má 12 vývodů, kde dle dokumentace 8 slouží pro ovládání segmentů a 4 pro selekci, které ze 4 číslic se mají rozsvítit [1]. Selekcce probíhá pomocí funkce `showNumber(int index)` a průběžně se během sekundy display stokrát překresluje, aby bylo možné zobrazit různé číslice na jednotlivých segmentech pomocí funkce `printNumber(int num, int dot)`. Vzhledem k tomu, že LED display podporuje i zobrazení desetinných čísel, tak jsem toho využil a zobrazené centimetry jsou v desetinných číslech. Na obrázku zobrazuje LED display hodnotu 3.785.

2 Měřič vzdálenosti HY-SRF05

Měřič má připojeno 5V z pinu Fitkitu 02 (opět pole P1) na vstup Ucc, a zem z pinu Fitkitu 49 na vstup GND. Dále má připojen výstupní pin Fitkitu 39 na vstup Trig a vstupní pin Fitkitu 40 na výstup Echo. Pin OUT měřiče není využit.

V dokumentaci ultrazvukového měřiče [2] jsem se dočetl, že k měření je potřeba první vyslat signál pomocí Trig a poté zapnout čítač a čekat až se signál vrátí na Echo. Poté, až se signál vrátí vypnout čítač a převést dobu na vzdálenost v cm podle vztahu:

$$Vzdálenost[cm] = Doba[\mu s]/58$$

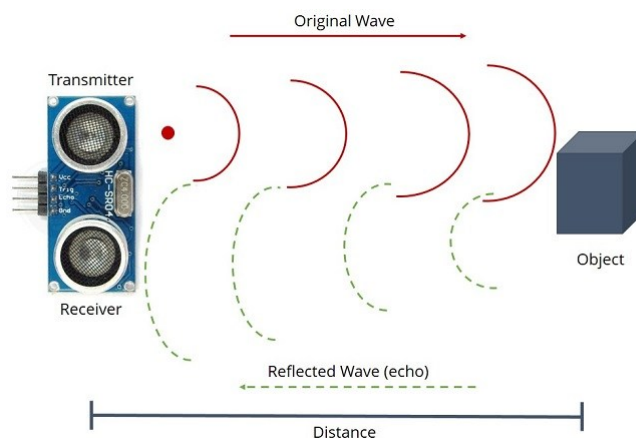
protože dle dokumentace [2][3] platí:

$$1cm = 58\mu s$$

3 Měření

Samotné měření probíhá tak, že vyslání signálu z pinu Trig (opět dle dokumentace [2]) probíhá $10\mu s$. Tj. doba po které tento měřič vyšle signál. Poté se opět výstup nastaví na 0 a vstoupí se do smyčky, ve které se čeká až na pin připojený na Echo přijde signál. To zjistíme tak, že na portu A, kam je Echo připojeno

nastavíme přerušení při sestupné hraně, a pull-up rezistor. V obsluze přerušení podle vztahu v sekci 2 spočítáme vzdálenost, zobrazíme na displayi a cyklus měření se opakuje.



Obrázek 3: Princip fungování měřiče

4 Časovač

4.1 LPTMR

LPTMR, tedy Low-Power Time časovač umožňuje generovat přerušení v nastaveném čase, čítá pulsy hodinového signálu, zpomalené předděličkou, přerušení je generováno v okamžiku dosažení přednastavené hodnoty. V okamžiku dosažení přednastavené hodnoty dokáže probudit MCU. Tento časovač se využívá pro překreslování displaye.

5 Závěr

Problém na který jsem narazil byl převod desetinného čísla na řetězec, který nešel pravděpodobně kvůli nedostatečnému výkonu vestavěného systému. Problém jsem poté vyřešil rozdělením desetinného čísla na dvě celá čísla, které už se lehce převedly na řetězec a mezi se vložila desetinná tečka. Problém mi taky dělalo vyznat se v poměrně komplexní dokumentaci použitého MCU. Nefunguje mi zobrazení kratší vzdálenosti než 9cm, mimo to výsledný program měří celkem korektní hodnoty a zobrazování probíhá správně.

Literatura

Reference

- [1] *LFD039AUE-102A-01, Zobrazovač LED : tme* [online]. [cit. 2018-11-25]. Dostupné z: <https://www.tme.eu/cz/details/lfd039aue-102a/ctyrmistne-led-displeje/wenrun/lfd039aue-102a-01/>
- [2] *HY-SRF05, ultrasonic : behnamrobotic* [online]. [cit. 2018-11-25]. Dostupné z: <http://dl.behnamrobotic.com/shop/datasheet/module/module-srf05-ultrasonic.pdf>
- [3] *K60 Sub-Family Reference Manual : freescale* [online]. [cit. 2018-11-25]. Dostupné z: http://cache.freescale.com/files/32bit/doc/ref_manual/K60P144M100SF2V2RM.pdf