

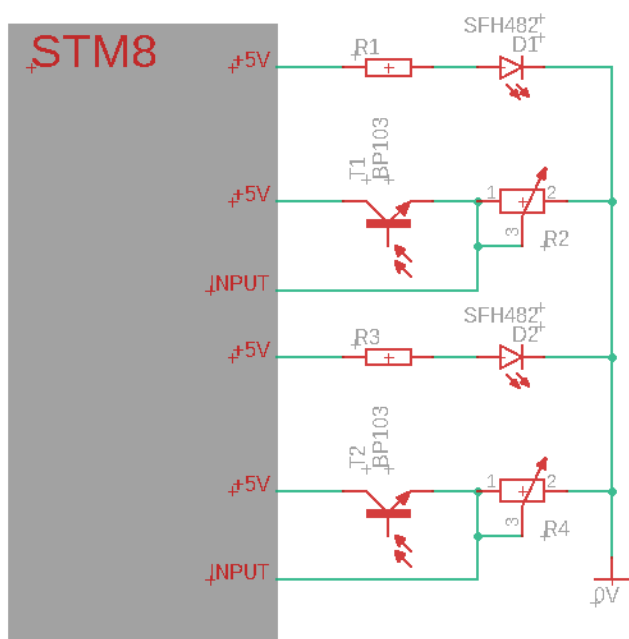
Dokumentace čítač průchodů

Popis Projektu

- Vytvořil jsem snímač průchodů který počítá kolik lidí prošlo dovnitř a pak od toho odečítá lidi kteří odešli ven
- Můj projekt obsahuje LCD displej, IR vysílače a přijímače, bzučák a komunikuje s počítačem pomocí UART

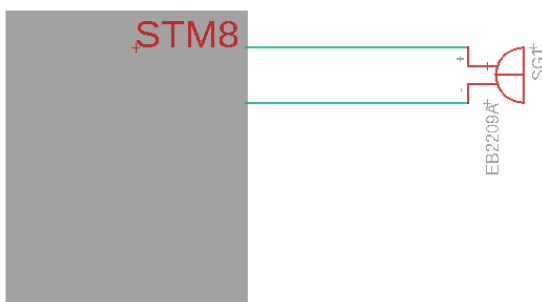
Optické brány

- Můj projekt obsahuje dvě optické brány postaveny hned za sebou, podle toho jaká se aktivuje první zjistí směr pohybu
- Optické brány jsou sestaveny pomocí IR Led a IR fototranzistoru



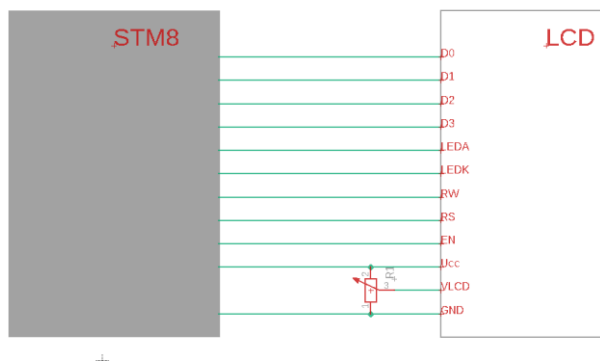
Bzučák

- Zapojení a použití pasivního bzučáku je velice jednoduché
- Můj bzučák má již v sobě vnitřní oscilátor takže do něj nemusím posílat vlastní frekvenci
- Jakmile na výstupu je přivedeno napětí tak začne vydávat zvuk o frekvenci určenou vnitřním oscilátorem

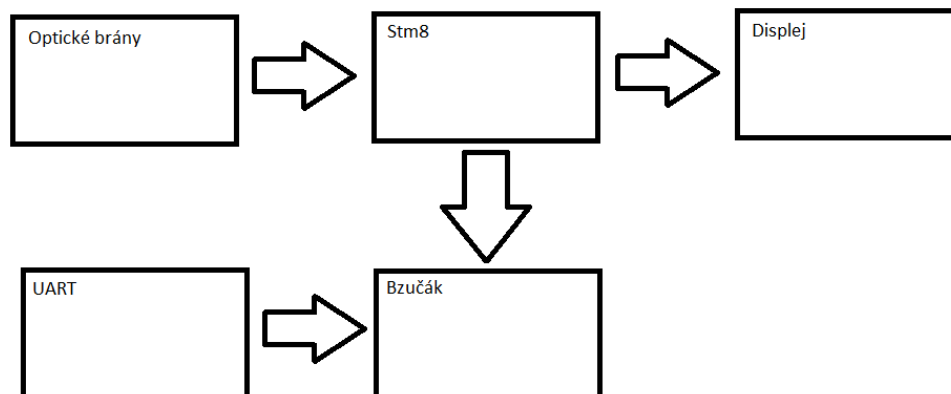


LCD Displej

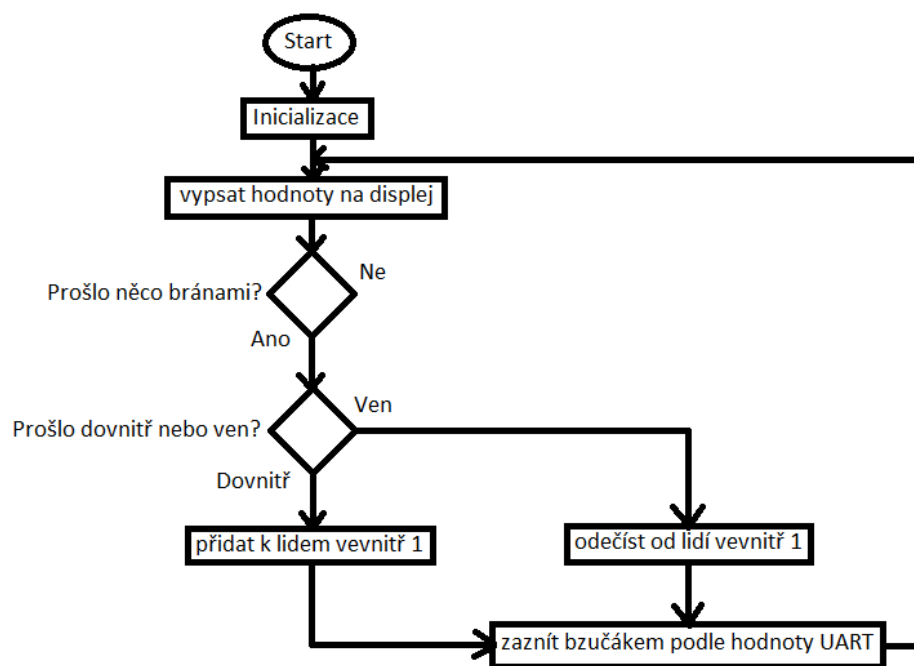
- Displej má šestnáct pinů - Datové piny :D0;D1;D2;D3;D4; D5; D6;D7 , Napájecí piny:VSS;VDD;VEE;LEDA;LEDK , Nastavovací piny Rs;Rw,E
- Můj displej má rozměr 16x2, takže jsou 2 sloupce a na každý se vejde 16 znaků



Blokové schéma



Vývojový diagram



Ukázka kodu

```
int8_t check_pruchod(void){
    uint32_t cas = 0;
    static int ready1=1;
    static int ready2=1;

    if (GPIO_ReadInputPin(IR_RX_1_PORT,IR_RX_1_PIN)!=RESET){
        ready1 = 1;
    }

    if (GPIO_ReadInputPin(IR_RX_2_PORT,IR_RX_2_PIN)!=RESET){
        ready2 = 1;
    }

    if (GPIO_ReadInputPin(IR_RX_1_PORT,IR_RX_1_PIN)==RESET && ready1 == 1){
        cas = millis();
        while ((millis() - cas)< 500){
            if (GPIO_ReadInputPin(IR_RX_2_PORT,IR_RX_2_PIN)==RESET){
                zvuk=1;
                ready1=0;
                return 1;
                delay_ms(100);
                break;
            }
        }
    }

    else if (GPIO_ReadInputPin(IR_RX_2_PORT,IR_RX_2_PIN)==RESET && ready2==1){
        cas = millis();
        while ((millis() - cas)< 500){
            if (GPIO_ReadInputPin(IR_RX_1_PORT,IR_RX_1_PIN)==RESET){
                zvuk=1;
                ready2=0;
            }
        }
    }
}
```

```

        return -1;
    delay_ms(100);
    break;
    }
    }
    }
    }

void main(void){
    char text[32];
    int8_t check = 0;
    int8_t check_zvuk = 0;
    int8_t pruchody = 0;
    uint32_t timeA = 0;
    uint32_t timeB = 0;
    uint32_t timeC = 0;

    setup();
    lcd_init();
    while (1){
if (GPIO_ReadInputPin(IR_RX_1_PORT,IR_RX_1_PIN)==RESET ||
    GPIO_ReadInputPin(IR_RX_2_PORT,IR_RX_2_PIN)==RESET){
        GPIO_WriteHigh(GPIOC,GPIO_PIN_5);
        }

        if ((milis() - timeA) > 70){
            timeA = milis();
            lcd_gotoxy(0 ,0);
            sprintf(text,"lidi uvnitr :%3d",pruchody+0);
            lcd_puts(text);
        }

        if ((milis() - timeB ) > 30){
            timeB = milis();
            check = check_pruchod();

```

```

        if (check!=1 && check != -1 && check != 0 ){

            check=0;

            }

        pruchody += check;

        if (pruchody < 0){

            pruchody=0;

            }

        }

    }

    if (zvuk == 1 && millis()-timeC > (100+speed*100)){

        timeC = millis();

        GPIO_WriteReverse(GPIOB,GPIO_PIN_2);

        check_zvuk ++;

        if (check_zvuk == 2){

            zvuk=0;

            check_zvuk=0;

            }

            }

            }

            }

    }

    INTERRUPT_HANDLER(UART1_RX_IRQHandler, 18)

    {

        char number;

        char c = UART1_ReceiveData8();

        UART1_SendData8(c);

        number= c - '0';

        if (number >=0 && number <=9){

            speed = number;

            }

            }

        char putchar (char c)

        {

```

```

        /* Write a character to the UART1 */
        UART1_SendData8(c);

        /* Loop until the end of transmission */
        while (UART1_GetFlagStatus(UART1_FLAG_TXE) == RESET);

        return (c);
    }

    char getchar (void)
    {
        char c = 0;
        while (UART1_GetFlagStatus(UART1_FLAG_RXNE) == RESET);

        c = UART1_ReceiveData8();

        return (c);
    }

```

Závěr

Sestavení Bzučáku

- Sestavení pasivního bzučáku je velice jednoduché takže mi to nedělalo problémy

Sestavení displeje

- Displej fyzicky sestavit nebylo nic těžkého, byl to spíše problém programování a to mi nedělalo problém
- S displejem sem si neobjednal patici takže sem dráty musel napájet přímo na desku

Sestavení optické brány

- Sestavení obvodu optické brány nebylo těžké
- Programovací část byla těžší kvůli tomu že jsem musel řešit spoustu problémů s tím kdy se má 1 přičítat a odečítat, co se stane když jsou oba přerušeny atd.
- Největší problém mi dělalo nalezení správné hodnoty odporu děliče napětí, bez něj by se obvod sám zastavoval kvůli IR světlu pozadí