PRINCÍPY POČÍTAČOV – HARDVÉR
CVIČENIE – ARDUINO

Časť 1: Programovanie Arduina, základná komunikácia

Arduino je mikrokontrolér, ktorý vie byť napájaný z USB, a komunikuje cez USB (pomocou integrovaného Serial-USB prevodníka). Na rozbehnutie komunikácie nám stačí Arduino, USB kábel, a PC.

Spustenie u seba:

1. Do PC nainštalujeme software Arduino IDE (stačí aj stará verzia 1.8.xxx)
2. Zapojíme Arduino cez USB do PC.

Pozn.: Niekedy je nutné nainštalovať driver (FTDI by malo fungovať, pre čínske klony je potrebné nájsť driver na Serial-USB prevodník CH340G/CH341G (CH341SER)

Spustenie v T3:

1. Zapneme Linux, otvoríme terminál a napíšeme "arduino"
2. Zapojíme Arduino cez USB do PC.
3. V Arduino IDE pod Tools: treba nastaviť Board (doska – Nano/Uno), procesor (Atmega328p, pre čínske klony Atmega 328p (old bootloader)), COM PORT (sériový port, ktorý zodpovedá arduinu), a bootloader: AVRISP mkII
4. Nahrajte minimalistický kód, ktorý bude blikať ledkou a vypisovať do sériového portu:

Kódy sú dostupné na: `davinci.fmph.uniba.sk/baranek14`

`01_basic_blink.ino`

```
void setup() {
  pinMode(LED_BUILTIN,OUTPUT);
  Serial.begin(9600);
}

void loop() {
  Serial.println("Hello World!");
  digitalWrite(LED_BUILTIN, HIGH);
  delay(1000);
  digitalWrite(LED_BUILTIN, LOW);
  delay(1000);
}
```

5. Nahrajte kód do arduina (ctrl+u). Nemal by vyhodiť žiadnu chybu. Arduino by malo začať blikať vstavanou ledkou. Po otvorení Serial Monitor-u (ctrl+m) a nastavení správneho baud rate (9600), by sa mali zobrazovať každé 2 sekundy nová správa *Hello World!* Ak je to tak, tak je všetko pripravené.
6. Stránka https://www.arduino.cc/reference/en/ obsahuje všetko potrebné.

Pozn: Užitočný je SerialMonitor (ctrl+m), ale aj Serial Plotter (pri analogRead).

Časť 2: Zapojenie Arduina do obvodu so zdrojom, Input/Output

Pozn.: Pre zníženie rizika poškodenia arduina používame 5V a 0V z laboratórneho zdroja, a Arduino iba prepojíme z GND breadboardu na GND pin. Pri prepájaní obvodov odpojíme USB kábel od Arduina, čo by ho malo ochrániť pred prípadným zlým zapojením. Po skontrolovaní zapojenia zapojíme arduino naspäť do USB a naprogramujeme ho.

Pre napájanie breadboardu použijeme pin Vin na arduine do +, GND do zeme. Počas prepájania káblov vypojte + z dosky.

Úloha: Modifikujte "počiatočný program", ktorý bliká vstavanou LED-kou, aby blikal LED-kou zapojenou v Breadboarde.

Riešenie:
Software: v programe treba prepísať "LED_BUILTIN" na ľubovoľný iný pin (napr. 3). Uploadnuť program.
Hardware: Treba zapojiť pin 3 na + vstup do rezistora a LEDky. Nezabudnúť prepojiť GND z mínusom.

Voliteľná úloha: Modifikujte program tak, aby LED-ka blikala rýchlejšie. Aký je limit, kedy pocitovo LED-ka prestane blikať? Čo sa stane, ak jeden delay zväčšíte a druhý zmenšíte?
Riešenie: meníme delay(), pri delay-i cca 50 (milisekúnd) už oko začne vidieť priemernú hodnotu. Menením časov nezávisle od seba dostávame rôzny jas LED-ky.

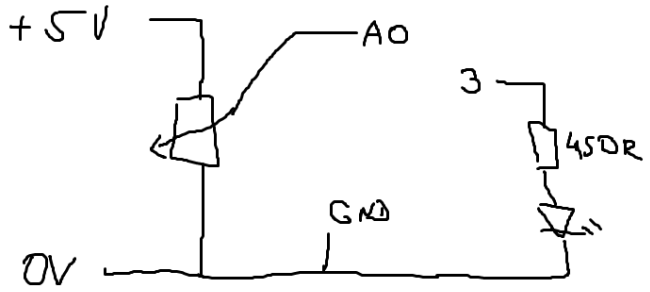
Asymetrický delay mení duty cycle – prešli sme do režimu Pulse Width Modulation. PWM sa používa takmer všade na hladké, takmer bezstratové riadenie výkonu (elektroautá, výťahy, displeje, nabíjačky, led osvetlenie,…). Táto funkcia je integrovaná do funkcie analogWrite na pinoch, ktoré majú možnosť PWM.

Úloha: Pomocou funkcie analogWrite zmeňte program tak, aby ledka menila svoju intenzitu medzi 3 rôznymi hodnotami.
Riešenie: digitalWrite(pin,HIGH) -> AnalogWrite(pin, integer hodnota). Pozor, musíte použiť pin s podporou PWM (má na sebe vlnovku).

Úloha: Nechajte LED zapojenú, na pin s analógovým vstupom pripojte stredný pin potenciometra. Upravte program tak, aby najprv: vypisoval hodnotu napätia na potenciometri do sériového portu. Upravte program tak, aby sa jas LEDky menil v závislosti od polohy potenciometra.

*Užitočná je funkcia map(a,b,c,d), ktorá premení premennú z rozsahu a-b do rozsahu c-d. Je to kvôli tomu, že analogRead berie hodnoty 0-1023, a analogWrite berie hodnoty 0-255. To isté sa dá docieliť x*255/1023. Pozn: Nie som si istý dátovými typmi.*

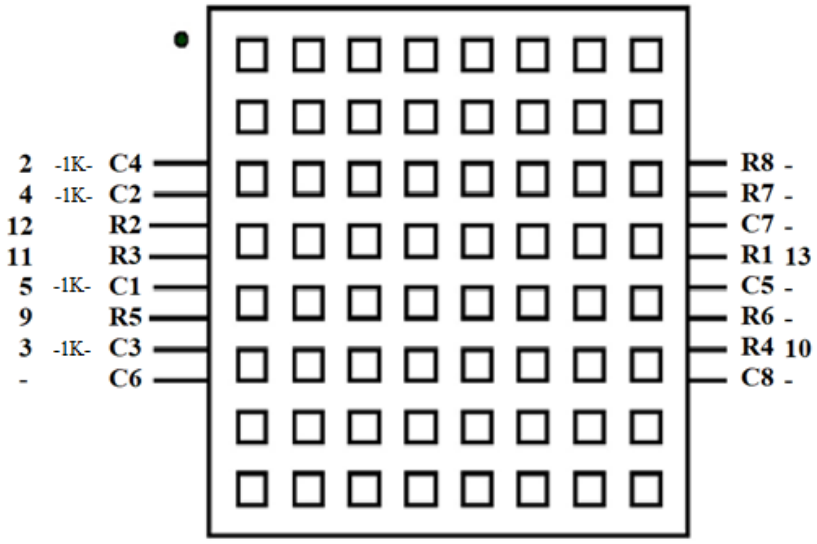
Riešenie:



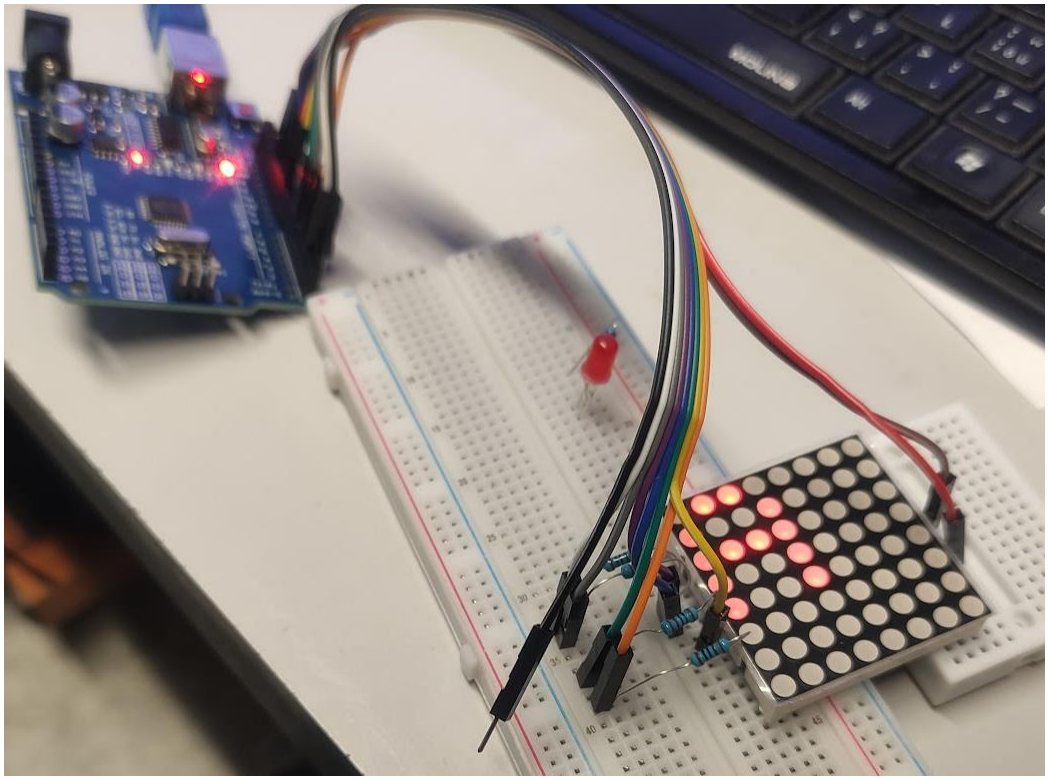
Výsledok sa dá taktiež stiahnuť ako 2. Úloha – Analógový zápis a čítanie.

Časť 3: 8x8 LED Matica

Zoberieme 8x8 Led maticu, a zapojíme z nej 5x4 LED maticu (stačí na jednoduché znaky). Poznámka: medzi C-čka a piny treba pridať 1K rezistor.



Stiahneme si program úloha 3 – LED matica. Zapojíme podľa schémy (ideálne nerozpájať zväzok káblikov), nezabudnúť na 1K rezistory a dať si pozor, aby linky bez rezistorov boli v správnom stlpci.



Malo by vám blikanie pomaly preskakovať po jednotlivých stĺpcoch a riadkoch, ako to ide v programe. Pre efekt “plného svietenia” zakomentujte (// - dve lomky na začiatku riadku) delay(50) vo for- loopoch, a hlavný delay(300) zmeňte napr. Na delay(3). Môžte skúsiť rôzne hodnoty.

Časť Z: I2C Multiplexer – zjednodušenie života

Zoberieme si 16x2 digitálny displej. Tento displej slúži na zobrazovanie ASCII znakov, 16 znakov v riadku, 2 riadky. Ak si všimnete "pinout" na displeji, má ich veľmi veľa. Na obsluhu displeja napriamo, cez jeho paralelné rozhranie, by bolo treba dosť veľa pinov z Arduina. Ak chceme spraviť nejaký komplexnejší projekt, obsahujúci viacero zariadení, piny nám môžu ľahko dôjsť.

Množstvo dát prenesených do displeja je však veľmi malé – na jeden "refresh" displeja nám stačí prenášať niekoľko desiatok bajtov. Preto, na redukciu počtu využitých pinov, môžme použiť tzv. Demultiplexer, a následne ho nejako rozumne ovládať. A práve tak je riešený modul k displeju 1602 – pomocou modulu PCF8574. Tento modul má 8 digitálnych výstupov, ktoré sa priamo zapoja do vstupov Displeja. Na komunikáciu s demux-om sa používa štandardný protokol vo svete mikrokontrolérov – IIC, alebo známejšie I^2C – "inter-integrated circuit". Je to synchrónna sériová komunikačná zbernica (master/slave), na krátke vzdialenosti. Používa dva vodiče – Serial Data (SDA) a Serial Clock (SCL). Môže sa ním paralelne zapojiť až 127 zariadení. Každé zariadenie má svoju 7-bitovú adresu (napr. Demux má adresu 0x20 alebo 0x27), na ktorej komunikuje. Každé zariadenie funguje buď ako kontrolér, alebo ako "cieľ" (slave, target). Štandardná rýchlosť je 100kbps.

Pre Arduino je komunikácia cez I2C spravená pomocou štandardnej knižnice "Wire", cez piny A4 (SDA) a A5 (SCL).

Úloha: zapojte I2C modul do displeja a do obvodu s Arduinom. Zoberte kód "I2C skener" a spustite ho, zistite adresu I2C modulu. (kód je v examples – wire – i2c scanner, alebo na webovej stránke.
Zapamätajte si adresu, zíde sa vám v ďalšej úlohe.

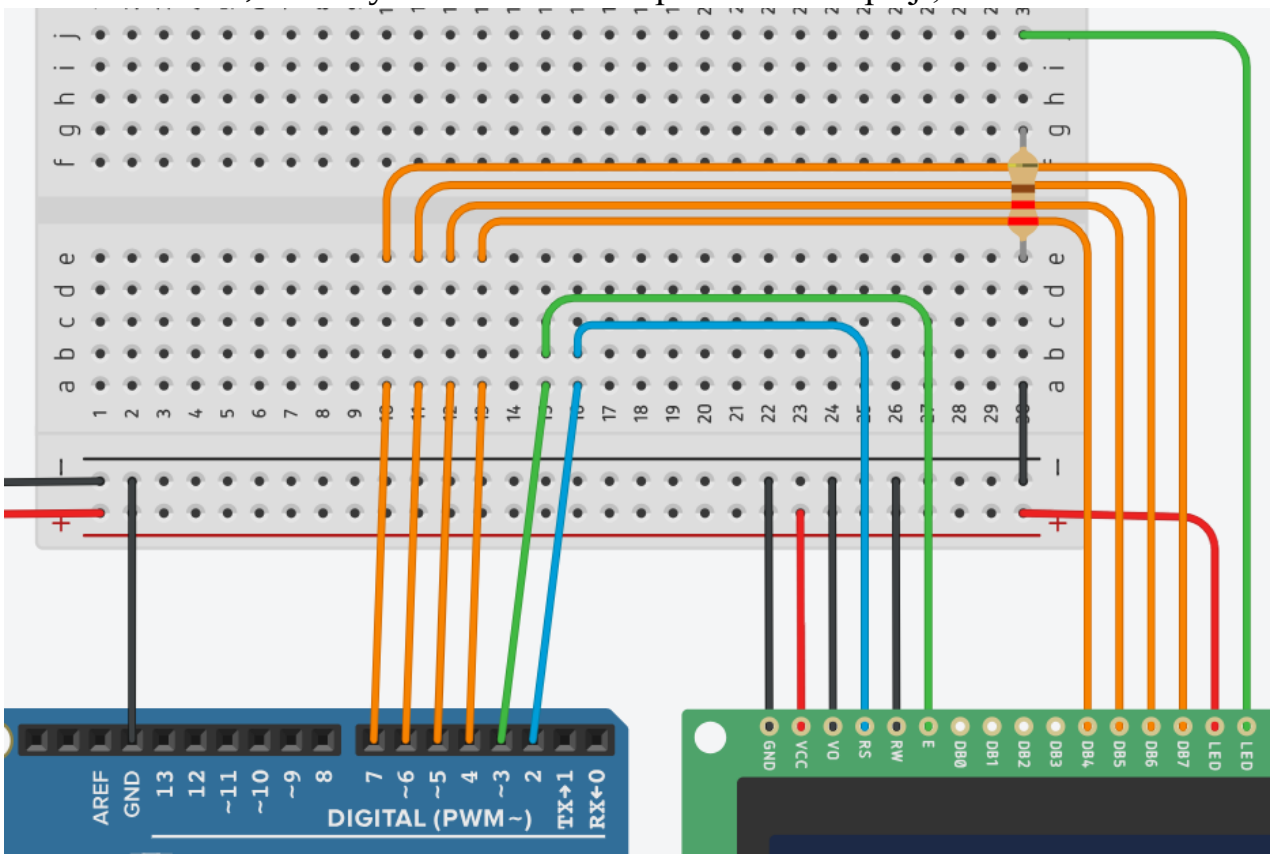
Výhoda Arduina je v množstve knižníc – nie je potrebné programovať priamo zapisovanie do demultiplexera, a vedieť hexadecimálne príkazy na ovládanie displeja. Stačí použiť knižnicu s dokumentáciou. Poznámka: Aktuálna knižnica LiquidCrystal I2C je na GitLabe, kde je prístup až po prihlásení, takže sa nedá úplne ľahko doklikať k príkazom.

Úloha: Použite knižnicu LiquidCrystal_I2C. Skontrolujte, či je stiahnutá v Arduino IDE, ak nie, použite library managera, stiahnte ju, nainštalujte, a pripojte sa k displeju na adrese, ktorú ste zistili v minulom kroku. Pozn: Knižníc je mnoho, do vyhľadávača v Library Managerovi zadajte " LiquidCrystal I2C", a nájdite knižnicu od "Marco Schwarz", mala by sa volať "Liquid Crystal I2C". Stiahnite si úlohu 5, a mala by vám vypísať "Hello world" na displej. Skúste zmeniť správu.

Časť Y: LCD displej – knižnice, “paralelný port”, a I2C protokol

ROBIŤ AŽ PO DOKÚPENÍ displeja 1602 bez I2C demultiplexera

Zoberieme si 16x2 digitálny displej. Tento displej slúži na zobrazovanie ASCII znakov, 16 znakov v riadku, 2 riadky. Ak si všimnete “pinout” na displeji, má ich veľmi veľa.



VCC/GND – napájanie LCD, VO – kontrast (stačí dať na GND), RS – register select (či zapisujeme príkaz alebo dáta), RW – read/write (zvyčajne iba zapisujeme; do GND), E – enable (či spracováva vstupy alebo nie) – ako CLK v klopných obvodoch, DB0-DB7 dátový bit 0 až 7 (dá sa použiť 4-bit mód, ako na obr, kde sa zapoja iba 4 piny).

Pre dočasné zjednodušenie života zapojíme schému ako na obrázku, a nahodíme tento kód nachádzajúci sa nižšie do arduina. Program obsahuje magickú formulku #include, ktorá importuje knižnicu LiquidCrystal.h na ovládanie práve týchto typov displejov

Program:

```
#include <LiquidCrystal.h>
// Parameters: (rs, enable, d4, d5, d6, d7)
LiquidCrystal lcd(2, 3, 4, 5, 6, 7);

void setup()
{
  lcd.begin(16, 2);
  lcd.clear();  // Print a message to the LCD.
  lcd.print(" Hello world!");
  lcd.setCursor(0, 1);
  lcd.print(" LCD Tutorial");
}
```

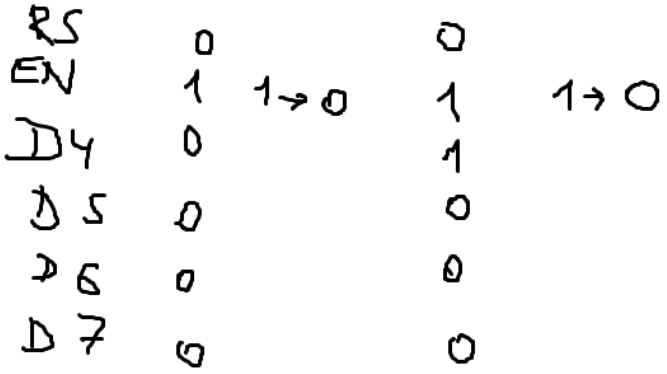
```
void loop()
{
  lcd.setCursor(15,1);
  lcd.print("-");
  delay(500);
  lcd.setCursor(15,1);
  lcd.print("/");
  delay(500);
}
```

Displej by mal fungovať, vpravo by sa mal otáčať – na / a naspäť. Na porozumenie, ako funguje knižnica však je potrebné použiť "Datasheet" – dokumentáciu k displeju, ktorý má v sebe opäť nejaký mikrokontrolér.

Úloha: Doprogramujte do programu v Arduine (časť setup) vymazanie obrazovky po tom, ako zobrazí LCD tutorial. Na túto časť nepoužite príkaz lcd.clear(), ale spravíte to manuálne, pomocou príkazov digitalWrite(), ktorými viete ovládať jednotlivé vstupy do mikrokontroléra.

Keďže ideme posielať iba "Command" a nie "Data", pin RS treba dať na 0, a napísať do D0-D7 príslušný príkaz (0x01), LSB first. Displej "prečíta" hodnoty na D0-D7 na hrane nadol na pine EN (High->Low). Ako si však môžte všimnúť používame iba horné 4 bity. To je úsporné opatrenie na šetrenie pinov. Preto najprv potrebujeme zapísať horné 4 bity (0x0 = 0b0000), poslať ich, a potom zapísať spodné 4 bity (0x1 = 0b0001).

Zápis príkazu 0x01:



```
void setup()
{
  lcd.begin(16, 2);// set up the LCD
  lcd.clear();
  lcd.print(" Hello world!");// Print a message to the LCD.
  lcd.setCursor(0, 1);// set the cursor to column 0, line 1
  lcd.print(" LCD Tutorial");
  delay(1000);
////////////////////////////////////////////////
  digitalWrite(EN_pin,HIGH); // prepare EN pin
  digitalWrite(RS_pin,LOW); // write command, not data - RS LOW

  //write 0x0 in binary = 0b0000
  digitalWrite(4,LOW);
  digitalWrite(5,LOW);
  digitalWrite(6,LOW);
  digitalWrite(7,LOW);
  delay(1);
  //make display read - high->low pulse
  digitalWrite(EN_pin,LOW);
  delay(1);
  digitalWrite(EN_pin,HIGH);
  delay(10);
  //write 0x1 0b0001
  digitalWrite(4,HIGH);
  digitalWrite(5,LOW);
  digitalWrite(6,LOW);
  digitalWrite(7,LOW);
  delay(1);
  //make display read - high-low pulse
  digitalWrite(EN_pin,LOW);
  delay(1);
  digitalWrite(EN_pin,HIGH);
}
```

Ak si dobre všimnete, tak vyššie máte napísaný príkaz lcd.clear(), ktorý v podstate robí presne to, čo ste krvopotne napísali na cca 20 riadkov.