# FIIT STU

# MIKRO cvičenie 4,5,6

# 

# 

# 

# 

# 

Meno: Jakub Gašparín, Dušan Havlík

Cvičenie: Utorok 18:00-19:40

Cvičiaci: Ing. Vladimír Kunštár

Obsah

[Úloha 1: 2](#_Toc120642643)

[Úloha 2: 4](#_Toc120642644)

[Úloha 3: 5](#_Toc120642645)

[Úloha 4: 8](#_Toc120642646)

[Úloha 5 16](#_Toc120642647)

[Úloha 6: 20](#_Toc120642648)

[Úloha 7: 23](#_Toc120642649)

[Úloha 8: 23](#_Toc120642650)

Úloha 1:  
Arduino

Obrázok, na ktorom je text

Automaticky generovaný popis

Sekcia 20: 8x LED diódy

Obrázok, na ktorom je text

Automaticky generovaný popis

4-miestny 7-segmentový LED display:

Obrázok, na ktorom je text, predajný automat

Automaticky generovaný popis

spínače log 1

Obrázok, na ktorom je text, červené, ovládací panel

Automaticky generovaný popis

spínače log 0

Obrázok, na ktorom je text, motor

Automaticky generovaný popis

klávesnica 4x4

Obrázok, na ktorom je text

Automaticky generovaný popis

LCD display, sekcia 11:

Obrázok, na ktorom je text, elektronika, obvod

Automaticky generovaný popis

# Úloha 2:



Musíme si pozrieť stranu 259:

I = U/R

R = 2,2 ohmov

I = 20 mA



const uint8\_t blink\_pins[] = { 2, 3, 4, 5, 6, 7, 8, 9 };

int pin\_index = -1;

int last\_pin\_index = 0;

void setup() {

  // put your setup code here, to run once:

  pinMode(2, OUTPUT);

  for (int i = 0; i < sizeof(blink\_pins)/sizeof(uint8\_t); i++)

  {

    pinMode(blink\_pins[i], OUTPUT);

    digitalWrite(blink\_pins[i], HIGH);

  }

  pin\_index = 0;

  last\_pin\_index = 0;

}

void loop() {

  // put your main code here, to run repeatedly:

  digitalWrite(blink\_pins[last\_pin\_index], HIGH);

  digitalWrite(blink\_pins[pin\_index], LOW);

  last\_pin\_index = pin\_index;

  pin\_index += 1;

  if (pin\_index >= sizeof(blink\_pins)/sizeof(uint8\_t))

    pin\_index = 0;

  delay(1000);

}

V tomto kóde sme si museli zadefinovať základné ledky a potom ich cez loop zapínať a vypínať.

# Úloha 3:

#define PA (2)

#define PB (3)

#define PC (4)

#define PD (5)

#define PE (6)

#define PF (7)

#define PG (8)

#define PDOT (9)

#define DIGIT\_PIN\_ROW (8)

const uint8\_t all\_pins[] = { PA, PB, PC, PD, PE, PF, PG, PDOT };

const int8\_t digit\_pins[] = {

  PG, PDOT, -1, -1, -1, -1, -1, -1, // 0

  PA, PD, PE, PF, PG, PDOT, -1, -1, // 1

  PC, PF, PDOT, -1, -1, -1, -1, -1, // 2

  PF, PE, PDOT, -1, -1, -1, -1, -1, // 3

  PA, PE, PD, PDOT, -1, -1, -1, -1, // 4

  PB, PE, PDOT, -1, -1, -1, -1, -1, // 5

  PB, -1, -1, -1, -1, -1, -1, -1, // 6

  PF, PG, PE, PD, PDOT, -1, -1, -1, // 7

  PDOT, -1, -1, -1, -1, -1, -1, -1, // 8

  PE, -1, -1, -1, -1, -1, -1, -1, // 9

};

void setup\_digit\_pins(void)

{

  for (int i = 0; i < sizeof(all\_pins)/sizeof(uint8\_t); i++) {

    pinMode(all\_pins[i], OUTPUT);

    digitalWrite(all\_pins[i], HIGH);

  }

}

void write\_digit(uint8\_t digit) {

  for (int i = 0; i < sizeof(all\_pins)/sizeof(uint8\_t); i++) {

    digitalWrite(all\_pins[i], LOW);

  }

  if (digit > 9)

    return;

  int8\_t \*pinptr = &digit\_pins[digit \* DIGIT\_PIN\_ROW];

  while (\*pinptr >= 0)

  {

    digitalWrite(\*pinptr, HIGH);

    pinptr++;

  }

}

void setup() {

  setup\_digit\_pins();

}

int index = 0;

void loop() {

  write\_digit(index);

  delay(1000);

  index += 1;

  if(index > 9)

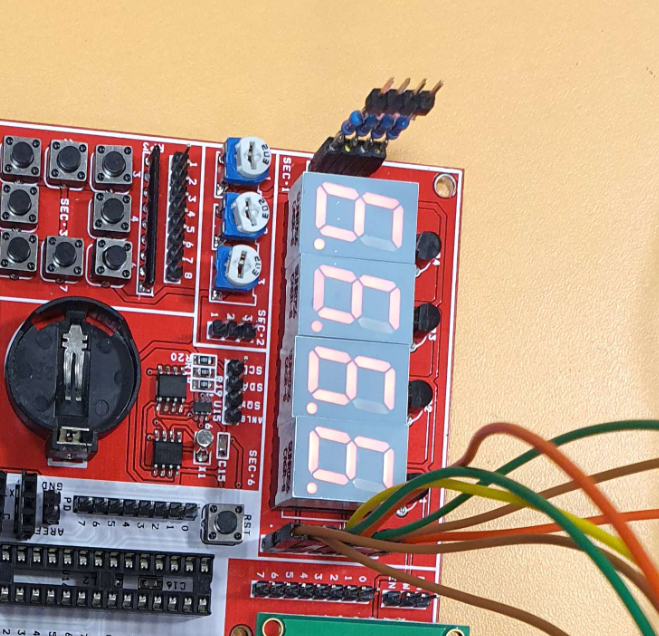
  {

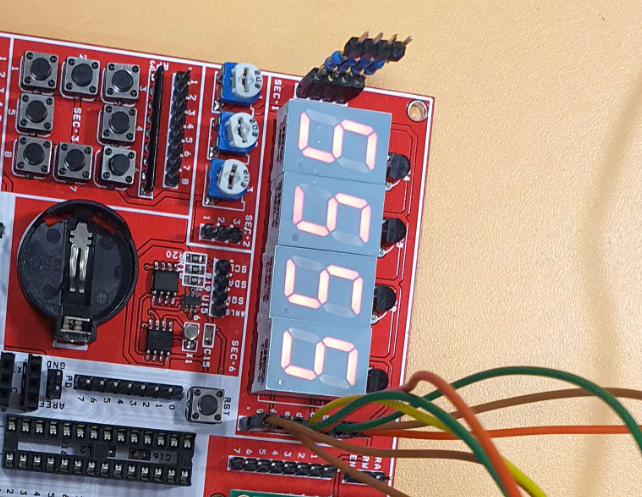
    index = 0;

  }

}

V tejto úlohe si bolo treba najprv. zadefinovať rôzne spustenia pre všetky cifry, každá cifra má iné vykreslenie na displeji. Ďalej sme cez for cyklus postupne vypísali všetky cifry na každej časti displeja jedna za druhou.





# Úloha 4:

#define PA (2)

#define PB (3)

#define PC (4)

#define PD (5)

#define PE (6)

#define PF (7)

#define PG (8)

#define PDOT (9)

#define PD1 (10)

#define PD2 (11)

#define PD3 (12)

#define PD4 (13)

#define DIGIT\_PIN\_ROW (8)

const uint8\_t all\_pins[] = { PA, PB, PC, PD, PE, PF, PG, PDOT };

const int8\_t digit\_pins[] = {

PG, PDOT, -1, -1, -1, -1, -1, -1, // 0

PA, PD, PE, PF, PG, PDOT, -1, -1, // 1

PC, PF, PDOT, -1, -1, -1, -1, -1, // 2

PF, PE, PDOT, -1, -1, -1, -1, -1, // 3

PA, PE, PD, PDOT, -1, -1, -1, -1, // 4

PB, PE, PDOT, -1, -1, -1, -1, -1, // 5

PB, -1, -1, -1, -1, -1, -1, -1, // 6

PF, PG, PE, PD, PDOT, -1, -1, -1, // 7

PDOT, -1, -1, -1, -1, -1, -1, -1, // 8

PE, -1, -1, -1, -1, -1, -1, -1, // 9

};

void setup\_digit\_pins(void)

{

for (int i = 0; i < sizeof(all\_pins)/sizeof(uint8\_t); i++) {

pinMode(all\_pins[i], OUTPUT);

digitalWrite(all\_pins[i], HIGH);

}

}

void write\_digit(uint8\_t digit) {

for (int i = 0; i < sizeof(all\_pins)/sizeof(uint8\_t); i++) {

digitalWrite(all\_pins[i], LOW);

}

if (digit > 9)

return;

int8\_t \*pinptr = &digit\_pins[digit \* DIGIT\_PIN\_ROW];

while (\*pinptr >= 0)

{

digitalWrite(\*pinptr, HIGH);

pinptr++;

}

}

String input\_string = "";

bool input\_complete = false;

void setup() {

setup\_digit\_pins();

pinMode(PD1, OUTPUT);

pinMode(PD2, OUTPUT);

pinMode(PD3, OUTPUT);

pinMode(PD4, OUTPUT);

Serial.begin(9600);

input\_string.reserve(256);

}

inline void select\_digit(uint8\_t i) {

if (i == 0) {

digitalWrite(PD1, LOW);

digitalWrite(PD2, HIGH);

digitalWrite(PD3, HIGH);

digitalWrite(PD4, HIGH);

} else if (i == 1) {

digitalWrite(PD1, HIGH);

digitalWrite(PD2, LOW);

digitalWrite(PD3, HIGH);

digitalWrite(PD4, HIGH);

} else if (i == 2) {

digitalWrite(PD1, HIGH);

digitalWrite(PD2, HIGH);

digitalWrite(PD3, LOW);

digitalWrite(PD4, HIGH);

} else if (i == 3) {

digitalWrite(PD1, HIGH);

digitalWrite(PD2, HIGH);

digitalWrite(PD3, HIGH);

digitalWrite(PD4, LOW);

}

}

void \_\_aasdserialEvent() {

while (Serial.available()) {

char recv = (char) Serial.read();

input\_string += recv;

if (recv == '\n') {

// do stuff

input\_complete = true;

}

}

}

uint16\_t frame\_timer = 0;

uint8\_t minuta = 59;

uint8\_t sekundy = 30;

void loop() {

uint8\_t cislo = minuta;

select\_digit(1);

write\_digit(cislo % 10);

digitalWrite(PDOT, LOW);

cislo /= 10;

delay(5);

select\_digit(0);

write\_digit(cislo % 10);

cislo /= 10;

delay(5);

cislo = sekundy;

select\_digit(3);

write\_digit(cislo % 10);

cislo /= 10;

delay(5);

select\_digit(2);

write\_digit(cislo % 10);

cislo /= 10;

delay(5);

// 20ms

frame\_timer += 20;

if (frame\_timer >= 1000) {

frame\_timer = 0;

sekundy += 1;

if (sekundy >= 60) {

sekundy = 0;

minuta += 1;

if (minuta >= 60) {

minuta = 0;

}

}

}

while (Serial.available()) {

char recv = (char) Serial.read();

input\_string += recv;

if (recv == '\n') {

uint8\_t target = 0;

uint8\_t i = 0;

uint8\_t hodnota = 10;

for (i = 0; i < input\_string.length(); i++) {

char c = input\_string.charAt(i);

if (c == ':' || c == '\n' || c == ' ')

break;

target += (c - '0') \* hodnota;

hodnota /= 10;

}

i += 1;

while (target >= 60)

target -= 60;

minuta = target;

target = 0;

hodnota = 10;

for (; i < input\_string.length(); i++) {

char c = input\_string.charAt(i);

if (c == '\n' || c == ' ')

break;

target += (c - '0') \* hodnota;

hodnota /= 10;

}

while (target >= 60) {

minuta += 1;

target -= 60;

}

sekundy = target;

frame\_timer = 0;

input\_string = "";

}

}

//if (input\_complete) {

// Serial.println(input\_string);

// input\_string = "";

// input\_complete = false;

//}

}

Obrázok, na ktorom je text, elektronika, obvod

Automaticky generovaný popis

# Úloha 5

const uint8\_t button\_pins[] = { 2, 3, 4, 5, 6, 7, 8, 9 };

uint8\_t last\_value = 0;

uint8\_t read\_button(uint8\_t which, uint8\_t accumulatedValue) {

uint8\_t pin = button\_pins[which];

uint8\_t mask = 1 << which;

uint8\_t newValue = 0;

uint8\_t currentValue = accumulatedValue;

uint8\_t button = digitalRead(pin) << which;

if ((currentValue & mask) != button) {

if ((currentValue & mask) == mask) {

// 1 -> 0

delay(3);

newValue = digitalRead(pin) << which;

if (newValue == 0) {

return accumulatedValue & ~mask;

}

} else {

// 0 -> 1

delay(8);

newValue = digitalRead(pin) << which;

if (newValue == mask) {

return accumulatedValue | mask;

}

}

}

return currentValue;

}

uint8\_t read\_buttons(void) {

uint8\_t value = 0;

value = read\_button(0, value);

value = read\_button(1, value);

value = read\_button(2, value);

value = read\_button(3, value);

value = read\_button(4, value);

value = read\_button(5, value);

value = read\_button(6, value);

value = read\_button(7, value);

return value;

}

void setup() {

pinMode(2, INPUT);

pinMode(3, INPUT);

pinMode(4, INPUT);

pinMode(5, INPUT);

pinMode(6, INPUT);

pinMode(7, INPUT);

pinMode(8, INPUT);

pinMode(9, INPUT);

Serial.begin(9600);

}

void loop() {

uint8\_t value = read\_buttons();

for (uint8\_t i = 0; i < 8; i++) {

uint8\_t mask = 1 << i;

if ((value & mask) != (last\_value & mask)) {

if ((value & mask) == mask) {

Serial.print("Stlacil som ");

Serial.println(i);

} else {

Serial.print("Pustil som ");

Serial.println(i);

}

}

}

if (last\_value != value) {

last\_value = value;

}

}Obrázok, na ktorom je text

Automaticky generovaný popis

# Úloha 6:

#define ISLOG1 (1)

const uint8\_t button\_pins[] = { 2, 3, 4, 5, 6, 7, 8, 9 };

uint8\_t last\_value = 0;

uint8\_t read\_button(uint8\_t which, uint8\_t accumulatedValue) {

uint8\_t pin = button\_pins[which];

uint8\_t mask = 1 << which;

uint8\_t newValue = 0;

uint8\_t currentValue = accumulatedValue;

uint8\_t button = digitalRead(pin) << which;

if ((currentValue & mask) != button) {

if ((currentValue & mask) == mask) {

// 1 -> 0

delay(ISLOG1 ? 8 : 3);

newValue = digitalRead(pin) << which;

if (newValue == 0) {

return accumulatedValue & ~mask;

}

} else {

// 0 -> 1

delay(ISLOG1 ? 3 : 8);

newValue = digitalRead(pin) << which;

if (newValue == mask) {

return accumulatedValue | mask;

}

}

}

return currentValue;

}

uint8\_t read\_buttons(void) {

uint8\_t value = 0;

value = read\_button(0, value);

value = read\_button(1, value);

value = read\_button(2, value);

value = read\_button(3, value);

value = read\_button(4, value);

value = read\_button(5, value);

value = read\_button(6, value);

value = read\_button(7, value);

if (ISLOG1) {

value ^= 255;

}

return value;

}

void setup() {

pinMode(2, INPUT);

pinMode(3, INPUT);

pinMode(4, INPUT);

pinMode(5, INPUT);

pinMode(6, INPUT);

pinMode(7, INPUT);

pinMode(8, INPUT);

pinMode(9, INPUT);

Serial.begin(9600);

}

void loop() {

uint8\_t value = read\_buttons();

for (uint8\_t i = 0; i < 8; i++) {

uint8\_t mask = 1 << i;

if ((value & mask) != (last\_value & mask)) {

if ((value & mask) == mask) {

Serial.print("Stlacil som ");

Serial.println(i);

} else {

Serial.print("Pustil som ");

Serial.println(i);

}

}

}

if (last\_value != value) {

last\_value = value;

}

}

Obrázok, na ktorom je text

Automaticky generovaný popis

# Úloha 7:

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXx

# Úloha 8:

#include <LiquidCrystal.h>

#define RA (2)

#define RW (3)

#define EN (4)

#define D0 (5)

#define D1 (6)

#define D2 (7)

#define D3 (8)

#define D4 (9)

#define D5 (10)

#define D6 (11)

#define D7 (12)

LiquidCrystal lcd(RA, RW, EN, D0, D1, D2, D3, D4, D5, D6, D7);

void setup() {

lcd.begin(16, 2);

lcd.clear();

}

void loop() {

lcd.setCursor(0, 0);

lcd.print("Dusan Havlik");

lcd.setCursor(0, 1);

lcd.print("Jakub Gasparin");

}



#include <LiquidCrystal.h>

#define RA (2)

#define RW (3)

#define EN (4)

#define D0 (5)

#define D1 (6)

#define D2 (7)

#define D3 (8)

#define D4 (9)

#define D5 (10)

#define D6 (11)

#define D7 (12)

LiquidCrystal lcd(RA, RW, EN, D0, D1, D2, D3, D4, D5, D6, D7);

uint8\_t cursor\_y = 0;

uint8\_t cursor\_x = 0;

void setup() {

// put your setup code here, to run once:

lcd.begin(16, 2);

lcd.clear();

Serial.begin(9600);

}

void print(char c) {

if (isprint(c)) {

if (cursor\_x >= 16) {

cursor\_x = 0;

cursor\_y++;

}

if (cursor\_y >= 2) {

lcd.clear();

cursor\_x = 0;

cursor\_y = 0;

}

lcd.setCursor(cursor\_x, cursor\_y);

lcd.write(c);

cursor\_x += 1;

}

}

uint8\_t control = 0;

void loop() {

// put your main code here, to run repeatedly:

while (Serial.available()) {

char c = (char) Serial.read();

if (control) {

if (c == 'c') {

lcd.clear();

cursor\_x = 0;

cursor\_y = 0;

}

control = 0;

continue;

}

if (c == '\\') {

control = 1;

continue;

}

print(c);

}

}

