

prog_Ardu_2024_12_16_2.txt

```
// Programy ETC22 2024_12_16_1 STM32Duino, pro ETC22

// PROGRAM 11
/* STM32G030J6 v SO8, kopirovani TL stisk LED1 na PA0 sviti,
nestisknute tlacitko LED1 nesviti */

# define LED1 PA0 // pin č. 1, PB5 SO8 STM32G030J6
# define TL PB7 // Tlacitko k GND pin č. 7, PB7 SO8 STM32G030J6

int tlacit=0; // promenna "tlacit" pro ulozeni stavu tlacitka
void setup() {

    pinMode(LED1, OUTPUT);
    pinMode(TL, INPUT_PULLUP); /* tlacitko a int. pull up rezistor */
}

void loop() {
    tlacit= ! digitalRead(TL); /* ulozit negovany stav na pinu TL do promenne "
tlacit" */
    digitalWrite (LED1, tlacit); /* nastavil LED ON , kdyz tlacitko je LOW */
    delay(100);
}

xxxxxxxxxxxxxxxxxxxxx
// PROGRAM 12

/* STM32G030J6 v SO8, kopirovani TL stisk, pak LED1 na PA0 sviti
a vypisuje na Terminal, stisk tlac = 1, nestisknute tlac = 0) */

# define LED1 PA0 // pin č. 1, PB5 SO8 STM32G030J6
# define TL PB7 // Tlacitko k GND pin č. 7, PB7 SO8 STM32G030J6

HardwareSerial Serial1 (PA10_R, PA9_R); // UART na remap. pinech č. 5 a 6

int tlacit=0; // promenna pro ulozeni stavu tlacitka

void setup() {
    Serial1.begin(9600);
    pinMode(LED1, OUTPUT);
    pinMode(TL, INPUT_PULLUP); /* tlacitko na PB7 s int. Pull UP rezistorem */
}

void loop() {
    tlacit= ! digitalRead(TL); // ulozit negovany stav na pinu TL
    digitalWrite (LED1, tlacit);
    delay(500);
    Serial1.print("tlacitko = ");
    Serial1.println(tlacit);
}

xxxxxxxxxxxxxxxxxxxxxxxxxxxxx
//PROGRAM 13 " prek OK"

// testuje tlacitko a vypisuje stav textove na terminal

/* STM32G030J6 v SO8, kopirovani TL stisk LED1 na PA0 svit
a vypisuje na Terminal, stisk tlac = 1, nestisknute tlac = 0) */

# define LED1 PA0 // pin č. 1, PB5 SO8 STM32G030J6
# define TL PB7 // Tlacitko k GND pin č. 7, PB7 SO8 STM32G030J6

HardwareSerial Serial1 (PA10_R, PA9_R); // UART na remap. pinech č. 5 a 6

int tlacit=0; // promenna pro ulozeni stavu tlacitka
```

```

void setup() {
  Serial1.begin(9600);
  pinMode(LED1, OUTPUT);
  pinMode(TL, INPUT_PULLUP);
}

void loop() {
  tlacit= ! digitalRead(TL); // ulozit negovany stav na pinu TL
  digitalWrite (LED1, tlacit);
  delay(500);
  Serial1.print("tlacitko ");
  if (tlacit == 1) {Serial1.println ("stisknuto "); }
  else { Serial1.println ("uvolneno "); }
}

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

// PROGRAM 14   "prek OK"

/* STM32G030J6 v SO8, kopirovani TL stisk LED1 na PA0 svit
a vypisuje na Terminal, stisk tlac = 1, nestisknute tlac = 0) */
# define LED1 PA0 // pin č. 1, PB5 SO8 STM32G030J6
# define TL PB7 // Tlacitko k GND pin č. 7, PB7 SO8 STM32G030J6
HardwareSerial Serial1 (PA10_R, PA9_R); // UART na remap. pinech č. 5 a 6
int tlacit=0; // promenna pro ulozeni stavu tlacitka
int pocet =0;

void setup() {
  Serial1.begin(9600);
  pinMode(LED1, OUTPUT); /* modra LED na pinu č. 4 PA0 */
  pinMode(TL, INPUT_PULLUP); /*pin c.7, PB7 jako vstup a interni PULL - UP
rezistor */
}

void loop() {
  tlacit= ! digitalRead(TL); // ulozit negovany stav na pinu TL
  digitalWrite (LED1, tlacit);
  delay(500);
  pocet++;

  Serial1.print(" test tlacitka cislo = ");
  Serial1.print (pocet);
  if (tlacit == 1) {Serial1.println (" stisknuto "); }
  else { Serial1.println (" uvolneno "); }
}

XXXXXXXXXXXXXXXXXXXX

//PROGRAM 15 "prek OK"

/*Testuje tlacitko, vypisuje dobu a počet testu ve formě

test tlacitka cislo = 36 cas = 18000 uvolneno
test tlacitka cislo = 37 cas = 18500 uvolneno
test tlacitka cislo = 38 cas = 19000 uvolneno
test tlacitka cislo = 39 cas = 19500 uvolneno */

/* STM32G030J6 v SO8, kopirovani TL stisk LED1 na PA0 svit
a vypisuje na Terminal, stisk tlac = 1, nestisknute tlac = 0) */
# define LED1 PA0 // pin č. 1, PB5 SO8 STM32G030J6
# define TL PB7 // Tlacitko k GND pin č. 7, PB7 SO8 STM32G030J6
HardwareSerial Serial1 (PA10_R, PA9_R); // UART na remap. pinech č. 5 a 6
int tlacit=0; // promenna pro ulozeni stavu tlacitka
int pocet =0;
unsigned long cas =0;

void setup() {

```

```

Serial1.begin(9600);
pinMode(LED1, OUTPUT);
pinMode(TL, INPUT_PULLUP);
}

void loop() {
  tlacit= ! digitalRead(TL); // uložit negovaný stav na pinu TL
  digitalWrite (LED1, tlacit);
  delay(500);
  pocet++;
  cas = millis();
  Serial1.print(" test tlacitka cislo = ");
  Serial1.print (pocet);
  Serial1.print(" cas = ");
  Serial1.print (cas);
  if (tlacit == 1) {Serial1.println (" stisknuto "); }
  else { Serial1.println (" uvolneno "); }
}

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
//Program 17 "prek OK"

/*tester reakce LED
na začátku červená LED zabliká, pak stisknout TL ,
rozsvítí se modrá LED1, čeká až zhasne modrá LED1 a pak hned uvolnit tlačítko.
Doba reakce je čas od zhasnutí LED1 do uvolnění tlačítka

není ošetřeno předčasné uvolnění tlačítka a další stavy
Namety pro práci studentu.
ošetřit předčasné uvolnění tlačítka před zhasnutím LED
je konstantní doba, za jak dlouho LED1 zhasne, zkusit modifikovat,
aby se doba, když LED1 zhasne změnila
přidat zvukový výstup podle rychlosti reakce,
doplnit limit času reakce
Aby se změnil jas LED ( červené), musí se epak odstranit rezistor pro BOOT,
který je mezi pinem c.8 a napájením VDD= +3,3 V */

/* tester reakce, STM32G030J6 v S08,
zabliká červená LED na pinu c. 8 pak přejde na nízký jas ,
stisknou tlačítko, rozsvítí modrá LED 1 , až zhasne , uvolnit tlačítko
podle reakční doby - OK zabliká 2x pomalu a párkrát ,
pomalejší reakce - zabliká 8x rychle a párkrát dlouze
vypíše na terminal reakční dobu

přidat Buzzer, akustickou signalizaci, upraví na kontrolu předčasného
uvolnění tlačítka,vypsát případně i předčasné uvolnění
dat modifikaci času , za jak dlouho LED zhasne
tak aby se je třeba optimalizovat frekvenci podle použitého buzzeru, aby to bylo
slyšet

*/

# define LED1 PA0 // pin č. 4, PB5 S08 STM32G030J6
# define LED PB5 // pin č. 8 PB5
# define TL PB7 // Tlačítko k GND pin č. 7, PB7 S08 STM32G030J6
# define BUZ PA13 // pin č. 7, PA13 S08 STM32G030J6
HardwareSerial Serial1 (PA10_R, PA9_R); // UART na remap. pinech č. 5 a 6

int tlacit=0; // proměnná pro uložení stavu tlačítka
int pocet =0;
unsigned long cas =0;
unsigned long mezicas =0;
unsigned long timeout =0;

```

```

unsigned long reakce =0;
unsigned long reakce1 = 1000;
unsigned long reakcemax = 500;

void setup() {
  Serial1.begin(9600);
  pinMode(LED, OUTPUT);
  pinMode(LED1, OUTPUT);
  pinMode(BUZ, OUTPUT);
  pinMode(TL, INPUT_PULLUP);
  tone(BUZ, 3000, 500); /* Buzzer pipne 2000 Hz , 500 ms dlouho */
}

void loop() {

  zablikej (LED, 5 , 200) ; // červená LED zablika
  analogWrite (LED, 20); // červená LED sniží jas

  while (! digitalRead(TL)==0) /* dokud není stisknuta tlačítka, nedelej nic */
  { // tedy je to nic nedělati

  }

  digitalWrite (LED1, HIGH); // Modrá LED ON
  delay (2000);
  /* pro kontrolu předčasného uvolnění tlačítka by se sem dala kontrola
  a hlášení chyby, studenti zkouší doplnit */
  digitalWrite (LED1, LOW); // Modrá LED OFF
  mezicas = millis();
  while (! digitalRead(TL)==1)
  {
  }
  cas= millis();
  reakce = cas - mezicas ; //
  Serial1.print(" reakce = ");
  Serial1.print (reakce);
  Serial1.println (" milisekund ");
  delay (1000);
  if (reakce <= 350)
  { tone(BUZ, 3500, 400); // rychlá reakce- kratké pipnutí
    zablikej (LED1, 2 , 600) ;

  }

  else {zablikej (LED1, 8 , 100) ;
  tone(BUZ, 4000, 2000); // pomalá reakce -dlouhé pipnutí
  }
}

void zablikej (int ledka, int kolikrat, int doba) { // naše funkce pro
zablikání
while ((kolikrat) >0) {
pinMode(ledka, OUTPUT);/* znovu definujeme výstupní mód pro zrušení "
analogového" módu */
digitalWrite (ledka, HIGH); // blikni LED
delay(doba);
digitalWrite (ledka, LOW); // zhasni LED
delay(doba);
kolikrat --;
}
}

```

prog_Ardu_2024_12_16_2.txt

xx
xx