

prog_Ardu_2024_12_16.txt

```
// Programy ETC22 2024 12 16 STM32Duino,

// PROGRAM 11
/* STM32G030J6 v S08, kopirovani TL stisk LED1 na PA0 svit */
# define LED1 PA0 // pin č. 1, PB5 S08 STM32G030J6
# define TL PB7 // Tlacitko k GND pin č. 7, PB7 S08 STM32G030J6

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

  pinMode(LED1, OUTPUT);
  pinMode(TL, INPUT_PULLUP);
}

void loop() {
  tlacit= ! digitalRead(TL); // ulozit negovany stav na pinu TL
  digitalWrite (LED1, tlacit);
  delay(100);
}

XXXXXXXXXXXXXXXXXXXX
// PROGRAM 12

/* STM32G030J6 v S08, kopirovani TL stisk LED1 na PA0 svit
a vypisuje na Terminal, stisk tlac = 1, nestisknute tlac = 0) */
# define LED1 PA0 // pin č. 1, PB5 S08 STM32G030J6
# define TL PB7 // Tlacitko k GND pin č. 7, PB7 S08 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 = ");
  Serial1.println(tlacit);
}

XXXXXXXXXXXXXXXXXXXXXXXXXXXX
//PROGRAM 13

// testuje tlacitko a vypisuje stav textove na terminal

/* STM32G030J6 v S08, kopirovani TL stisk LED1 na PA0 svit
a vypisuje na Terminal, stisk tlac = 1, nestisknute tlac = 0) */
# define LED1 PA0 // pin č. 1, PB5 S08 STM32G030J6
# define TL PB7 // Tlacitko k GND pin č. 7, PB7 S08 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
```

```

                                prog_Ardu_2024_12_16.txt
digitalWrite (LED1, tlacit);
delay(500);
Serial1.print("tlacitko ");
if (tlacit == 1) {Serial1.println ("stisknuto "); }
else { Serial1.println ("uvolneno "); }
}

```

XX

PROGRAM 14

```

/* 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);
  pinMode(TL, INPUT_PULLUP);
}

```

```

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

```

/*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); // ulozit negovany stav na pinu TL
  digitalWrite (LED1, tlacit);
  delay(500);
  pocet++;
}

```

prog_Ardu_2024_12_16.txt

```
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 "); }
}

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
/*PROGRAM 16 netestováno

pracovni pro test reakce zatím LED a LED1 - zablika po stisku */

/* 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. 4, PB5 SO8 STM32G030J6
# define LED PB5 // pin č. 8 PB5
# define TL PB7 // Tlacitko k GND pin č. 7, PB7 SO8 STM32G030J6
# define BUZ PA13 // pin č. 7, PA13 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(LED, OUTPUT);
  pinMode(LED1, OUTPUT);
  pinMode(BUZ, OUTPUT);
  pinMode(TL, INPUT_PULLUP);
}

void loop() {
  zablikej (LED, 5 , 100) ;
  while (! digitalRead(TL)==0)
  {
  }
  zablikej (LED1, 8, 100) ;

  /*tlacit= ! digitalRead(TL); // ulozit negovany 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 "); } */
}
/* start zablikat dobu LED
stiknout tlacitko
rozsvitit LED1
po stisku v nějakou dobu zhasnout LED1 a merit dobu reakce a vypisovat, bliknou

LED , kdyz v limitu, zablikat vicerktrak LED1 ,
kdyz mimolimit
a pokračovat
*/
void zablikej (int ledka, int kolikrat, int doba) { // nase funkce pro
zablikani
while ((kolikrat) >0) {
digitalWrite (ledka, HIGH); // blikni LED
delay(doba);
```

```

prog_Ardu_2024_12_16.txt
digitalWrite (ledka, LOW); // zhasni LED
delay(doba);
kolikrat --;
Serial1.print (kolikrat);
}
}

```

```

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
//Program 17

```

```

/*tester reakce LED
na zacatku cervena LED zablika, pak stisknou TL ,
rozsviti se modra LED1, cekat az zhasne modra LED1 a pak hned uvolnit tlacitko.
neni osetreno predcasne uvolneni tlacitka a dalsi-
pro praci studentu.
je konstantni doba, za jak dloho LED1 zhasne, studenti- modifikovat, aby se tato
doba menila.

```

```

/* tester reakce, STM32G030J6 v S08,
zablika cervena LED na pinu c. 8 pak prejde na nizky jas ,
stiknou tlacitko, rozsviti modra LED 1 , az zhasne, uvolnit tlacitko
podle reakcni doby - OK zablika 2x pomalu, spatne zablika rychle
vypise na terminal reakcni dobu

```

```

pridat Buzzer, akustickou signalizaci, upravi na kontrolu predcasneho
uvolneni tlacitka,vypsati pripadne i predcasne uvolneni
dat modifikaci casu , za jak dlouho LED zhasne
pole

```

```

*/
# define LED1 PA0 // pin č. 4, PB5 S08 STM32G030J6
# define LED PB5 // pin č. 8 PB5
# define TL PB7 // Tlacitko 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; // promenna pro ulozeni stavu tlacitka
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);
}

```

```

void loop() {
  zablikej (LED, 5 , 200) ; // cervena LED zablika
  analogWrite (LED, 20); // cervena LED snizi jas

```

```

//
while (! digitalRead(TL)==0)

```

```
{
}

digitalWrite (LED1, HIGH); // Modra LED ON
delay (2000);
digitalWrite (LED1, LOW); // Modra 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 <= 300)
{ zablikej (LED1, 2 , 600) ;}

else {zablikej (LED1, 8 , 100) ;}
}

void zablikej (int ledka, int kolikrat, int doba) { // nase funkce pro
zablikani
while ((kolikrat) >0) {
pinMode(ledka, OUTPUT);// znovu definujeme vystupni mod pro zrušení "
analogoveho"

modu
digitalWrite (ledka, HIGH); // blikni LED
delay(doba);
digitalWrite (ledka, LOW); // zhasni LED
delay(doba);
kolikrat --;
}
}

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```