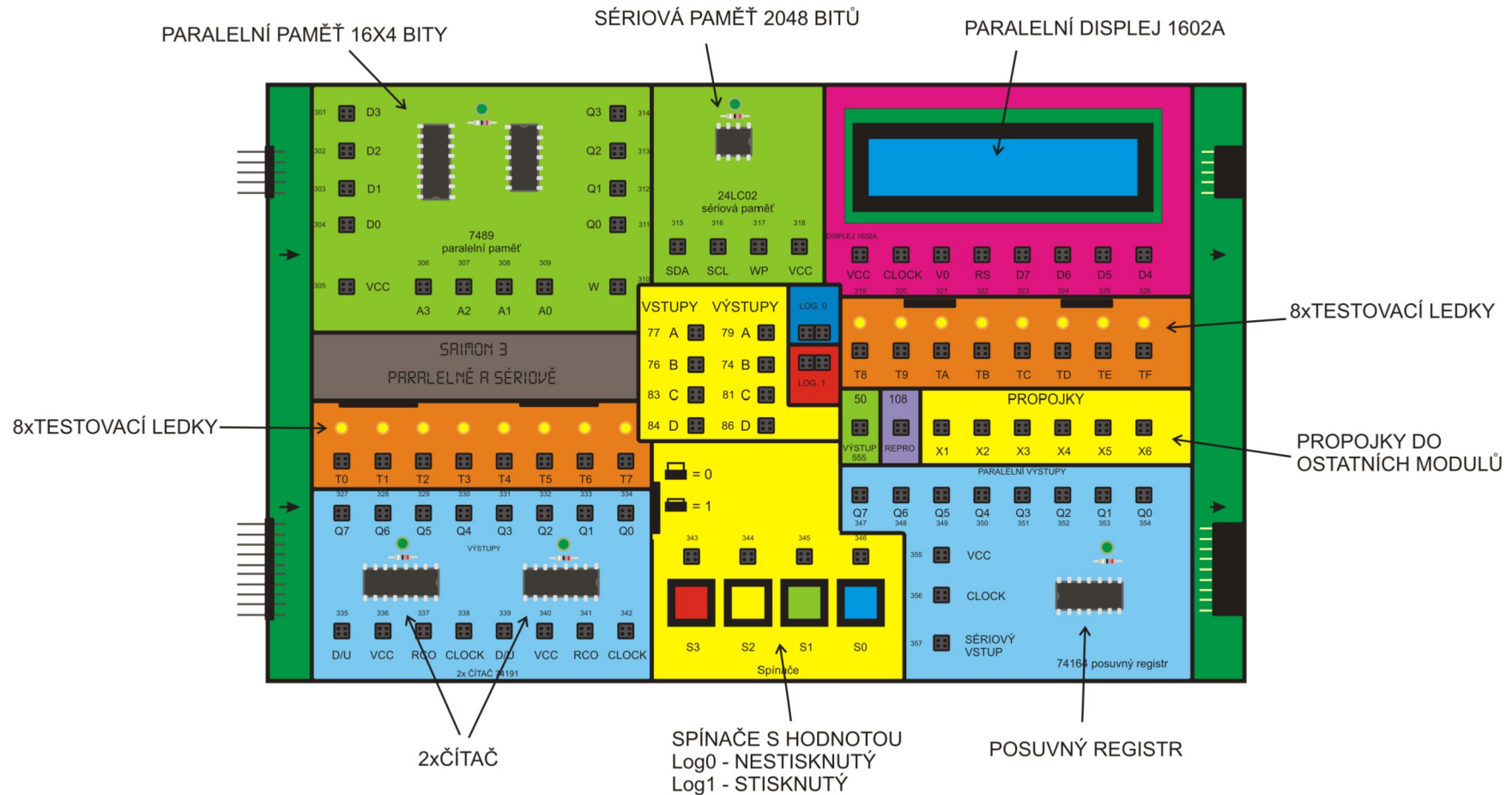


# SAIMON 3

NÁVOD - ČÁST PRVNÍ  
OBVODY NA SAIMON 3  
ANEB CO TO UMÍ

## Úvod

Tato první část návodu pro modul Saimon 3 bude obsahovat pouze jednoduchý popis obvodů na modulu a jejich základní funkci. Každému obvodu bude věnovaná jedna kapitola. Teprve druhá část návodu popíše jejich použití pro tvorbu SSO a také programování (pro majitele Saimon 4 – Arduino). Saimon 3 obsahuje následující obvody:



Obrázek 0.1 – Obvody na modulu Saimon 3

## **Kapitola 1 – Paralelní paměť 16x4 bity**

Paralelní paměť je tvořena dvěma integrovanými obvody. Vlastní paměť – IO 7489 a nám již známý obvod 7400, který obsahuje 4 logické funkce NAND. Proč je paměť tvořena dvěma obvody, když by mohl stačit jen jeden, vysvětlím na konci kapitoly.

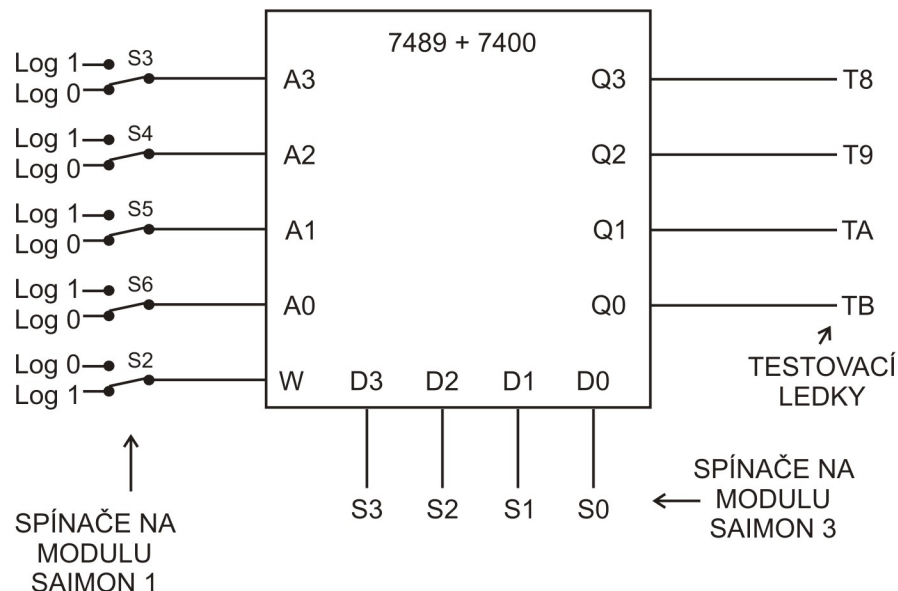
Co to tedy je paměť 16x4 bity? Vzpomeňte si na obvod 74175 modulu Saimon 1. To byla také paměť. Její funkce byla zapamatovat si stav vstupů. Tedy pokud jsme na vstupy A,B,C,D připojily kombinaci hodnot log0 a log1, pak přivedli náběžnou hranu na zdířku ZAPSAT, tak se odpovídající výstupy A,B,C,D nastavily na stejné hodnoty, jaké měly vstupy. Pozdější změna hodnot vstupů už neměla vliv na hodnoty výstupů. Hodnoty vstupů se zapsaly do paměti. Jaké hodnoty jsou v paměti zapsané pak bylo možné zjistit na výstupech. Kapacita paměti 74175 byla 4 bity (4 vstupy A,B,C,D).

Obvod 7489 funguje podobně. Stejným způsobem si zapamatuje vstupy. Vstupy jsou zde značené D0,D1,D2,D3 (namísto VSTUP A, VSTUP B, VSTUP C, VSTUP D) a výstupy jsou značené Q0, Q1, Q2, Q3 (namísto VÝSTUP A, VÝSTUP B, VÝSTUP C, VÝSTUP D). Zdířka ZAPSAT je zde značená „W“ (z anglického WRITE). Co je zde navíc, jsou zdířky A3,A2,A1,A0. Tyto zdířky znamenají **adresu** paměťové buňky (z anglického Address).

Obvod 74175 obsahoval 4 bity pro zapamatování hodnot. Tomu budu říkat 1 **paměťová buňka**. Představte si, že obvod 7489 uvnitř obsahuje 16 takových paměťových buněk a každá má velikost 4 bity. Abychom věděli, do které z šestnácti buněk chceme zapisovat (nebo z ní číst), musíme to obvodu 7489 nějak sdělit. A k tomu slouží zdířky A3,A2,A1,A0. Na ně přivedeme kombinaci bitů, která bude určovat, jakou paměťovou buňku budeme používat. Například kombinace A3=0,A2=0,A1=0,A0=0 bude znamenat zápis (nebo čtení) z buňky číslo 0. Jiná kombinace, například A3=1,A2=0,A1=0,A0=1 bude znamenat zápis (nebo čtení) z buňky číslo 9 (1001 je binární kombinace pro číslo 9).

## Jak sestavit obvod pro zápis do paměti?

Nejdříve schéma. To je na obrázku 1.1.

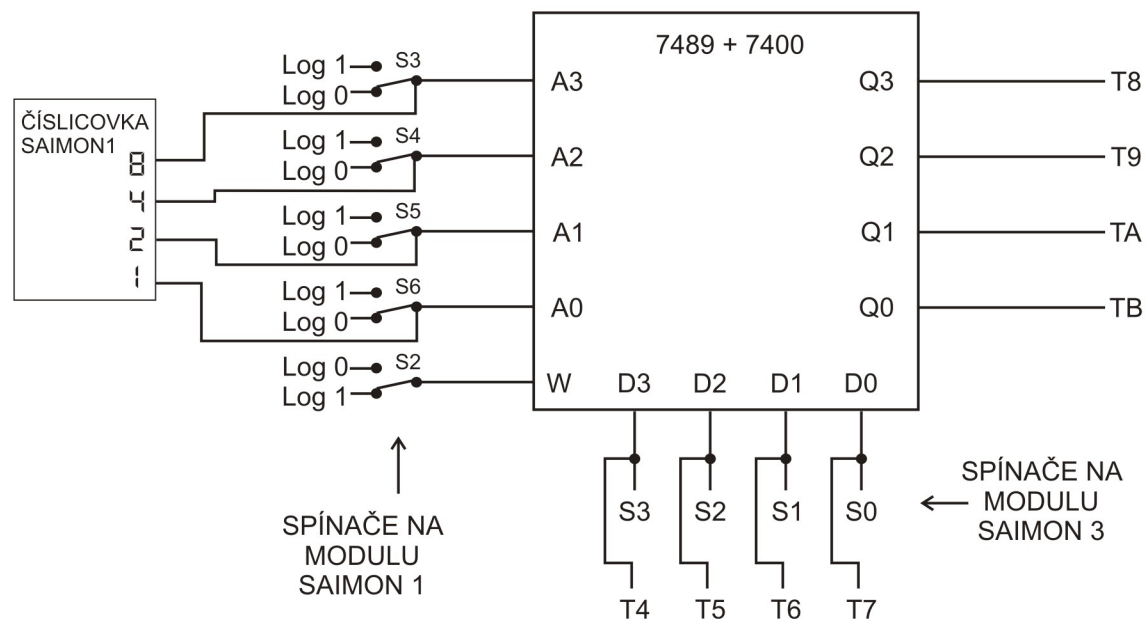


Obrázek 1.1 – Schéma obvodu pro zápis do paměti

Pomocí spínačů S3 až S6 na modulu Saimon 1 (použijte propojky X) určíme adresu paměťové buňky. Kombinací spínačů S3 až S0 na modulu Saimon 3 určíme hodnotu, kterou do buňky zapíšeme. Zápis provedeme stiskem spínače S2. Jakmile provedeme zápis, hodnoty vstupů se okamžitě zobrazí na výstupech. Rozsvítí se odpovídající testovací ledky.

**Pozor!** - může to být matoucí - adresy A3 až A0 jsou na Saimon 3 vodorovně dole pod paměťovým obvodem a datové vstupy D3 až D0 jsou svisle vlevo. Ve schématu na obrázku 1.1 jsou adresní zdířky svisle vlevo a datové zdířky vodorovně dole, stejně tak zdířka W je jinde. Jak už víme, blokové schéma většinou vypadá jinak, než reálný obvod.

Schéma můžeme vylepšit – Adresu budeme zobrazovat jako číslo. A hodnotu výstupů také jako číslo a zároveň jako kombinaci na testovacích ledkách. Zapisovanou hodnotu budeme také zobrazovat, a to na testovacích ledkách. Schéma vylepšeného zápisu je na obrázku 1.2.



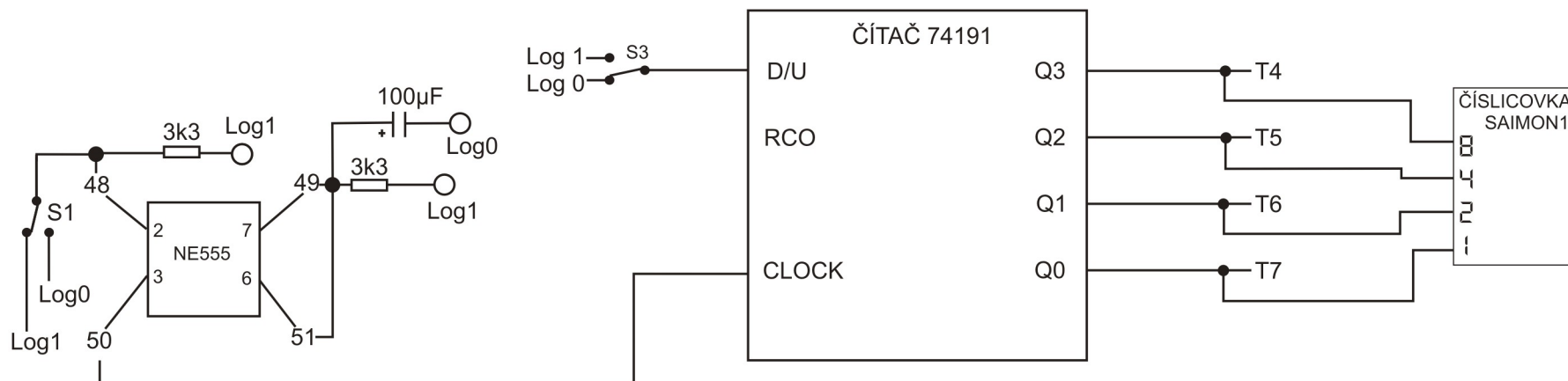
Obrázek 1.2 – Obvod pro zápis do paměti se zobrazením čísel

Číslíková na modulu Saimon 1 bude zobrazovat číslo, které bude znamenat adresu používané buňky. Hodnota uložená v této buňce bude zobrazena na testovacích ledkách T8,T9,TA,TB. Zapisovaná hodnota bude zobrazena na ledkách T4,T5,T6,T7. Po stisku spínače S2 se stejná kombinace objeví na ledkách T8,T9,TA,TB.

K paměti se v tomto dílu návodu ještě vrátíme. Teď vysvětlení, které jsem slíbil na začátku kapitoly – proč paměť tvoří dva obvody. 7489 je obvod pro samotnou paměť. Ten však zapamatované hodnoty z nějakého mne neznámého důvodu zobrazuje inverzně – tedy místo log0 zobrazuje log1 a místo log1 zobrazuje log 0. Potřeboval jsem tyto výstupní hodnoty ještě „otočit“. Výstupy jsou čtyři a k jejich otočení jsem potřeboval čtyři logické funkce NOT. A ty jsem sestavil ze čtyř logických funkcí NAND tak, jak jsme si ukazovali v návodu na Saimon 1. Teď obvod funguje tak, jak je logické – když připojím kombinaci hodnot log0 a log1 na vstupy a zapíšu do paměti, tak stejná kombinace se objeví na výstupech. Fotka obvodu je na obrázku 1.3.

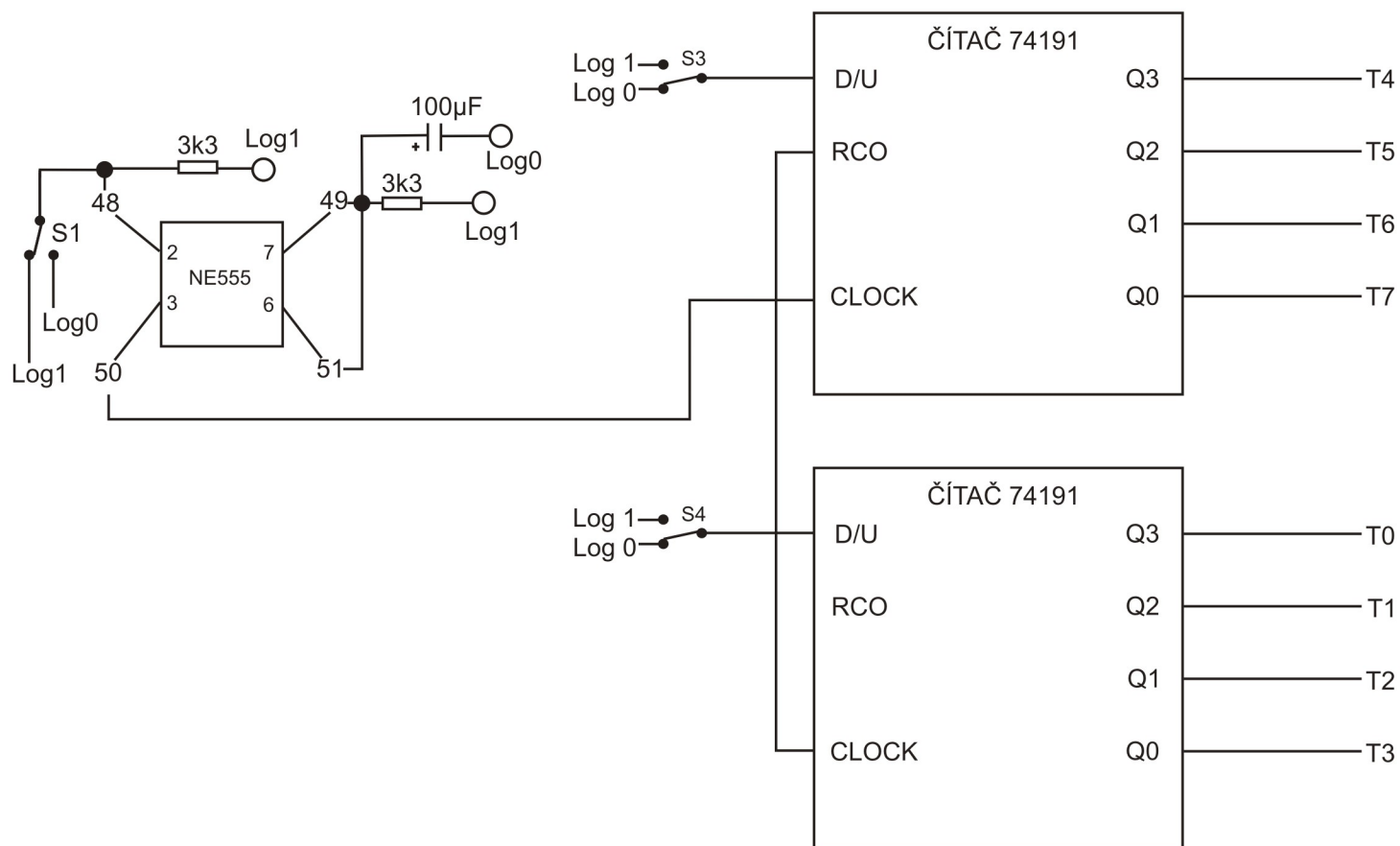
## Kapitola 2 – Čítač 74191

Čítač je obvod, který přičítá jedničku k číslu, které je v něm uložené. Čítač na modulu Saimon 3 lze nastavit i tak, aby odečítal. Přičítání nebo odečítání se děje v chvíli, kdy přivedeme hodinový pulz (náběžnou hranu) na vývod CLOCK. Pojďme se podívat na schéma.



Obrázek 2.1 – Zapojení čítače 74191

Na obrázku 2.1 je zapojen jeden ze dvou čítačů, které jsou na modulu Saimon 3. Číslo, které je v čítači uložené, tvoří čtyři bity. Ty jsou vyvedené na zdířky Q3, Q2, Q1, Q0. Hodnota čísla je tedy od 0 do 15 (resp. F). Na zdířku CLOCK jsem připojil monostabilní zapojení časovače NE555 a hodinové pulzy generuji stiskem spínače S1. Spínačem S3 mohu volit, zda bude čítač přičítat, nebo odečítat. Značení zdířky D/U je z anglického Down/Up – Dolu/Nahoru. Binární zápis čísla je zobrazen na testovacích ledkách T4, T5, T6, T7. Hexadecimální zápis čísla je vidět na číslicovce. Ve chvíli, kdy čítač napočítá číslo 15 a vygenerujeme další hodinový pulz, čítač změní hodnotu čísla na 0 a vygeneruje náběžnou hranu na zdířce RCO. Toho můžeme využít při zapojení dvou čítačů do série. Zapojení dvou čítačů je na obrázku 2.2.



Obrázek 2.2 – Zapojení dvou čítačů do série

Na obrázku 2.2 vidíme zapojení dvou čítačů. Na zdiřku CLOCK prvního čítače, který je zobrazen nahoře, je přiveden hodinový signál z časovače NE555. Čítač bude přičítat k uložené hodnotě čísla. Ve chvíli, kdy se číslo změní z 15 na 0, vygeneruje náběžnou hranu na zdiřce RCO (z angl. Ripple Carry Out). RCO prvního čítače je připojené na CLOCK druhého čítače (obvod dole na obrázku). Druhý čítač přičte jedničku ke svému číslu. Takto můžeme počítat od 0 do 255. U obou čítačů můžeme volit pomocí spínačů na zdiřkách D/U, zda budou přičítat, či odčítat.

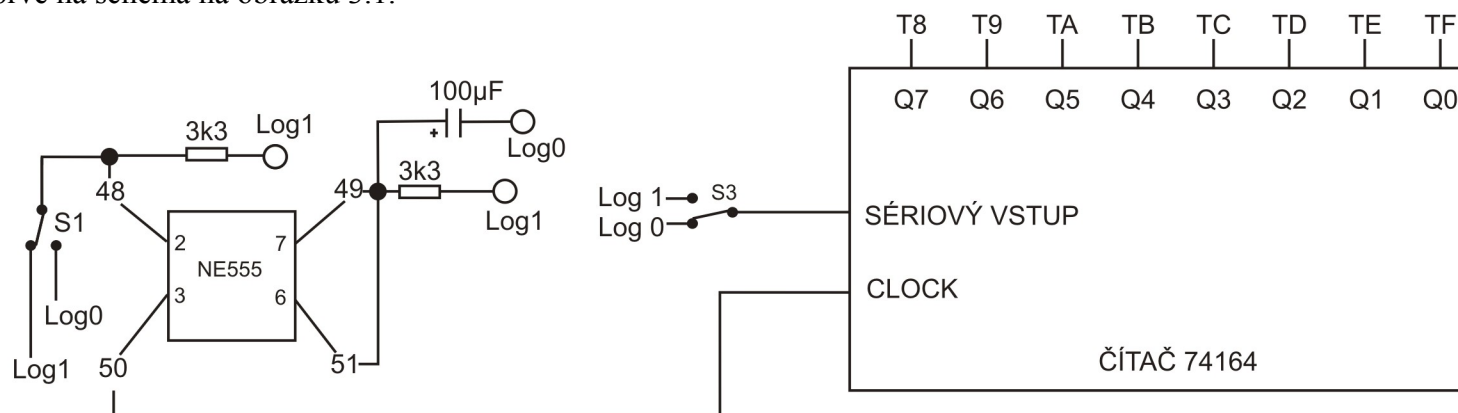
Čítač využijeme při navrhování SSO jako počítadlo stavů SSO. Také ho použijeme pro generování adresy pro paralelní paměť - obvod 7489.

### Kapitola 3 – Posuvný registr 74164

Posuvný registr je obvod, který umí převést sériový vstup na paralelní výstup. Slova sériový a paralelní známe už z návodu pro Saimon 1. Například při zapojování rezistorů a kondenzátorů jsme používali sériové a paralelní zapojení. Sériové zapojení rezistorů znamenalo, že rezistory byly v obvodu zapojeny postupně „za sebou“. Elektrický proud musel protéct nejprve prvním rezistorem a pak mohl protéct druhým. Paralelní zapojení rezistorů znamenalo, že rezistory byly v obvodu zapojeny „vedle sebe“. Elektrický proud protékal všemi rezistory zároveň.

S binárním číslem je to podobné. My jsme zatím všude používaly paralelní přenos binárního čísla. Číslicovka má čtyři paralelní vstupy. Všechny bity přivedeme na číslicovku ve stejný okamžik a zobrazí se číslo. Bity jsou přivedené na číslicovku neustále a ve chvíli, kdy jejich hodnoty změním, změní se i číslo.

Sériový přenos binárního čísla je něco, s čím jsme se ještě nesetkali. Bity jdou za sebou a musí nám něco signalizovat, kolikátý bit právě jde. Podívejme se nejprve na schéma na obrázku 3.1.

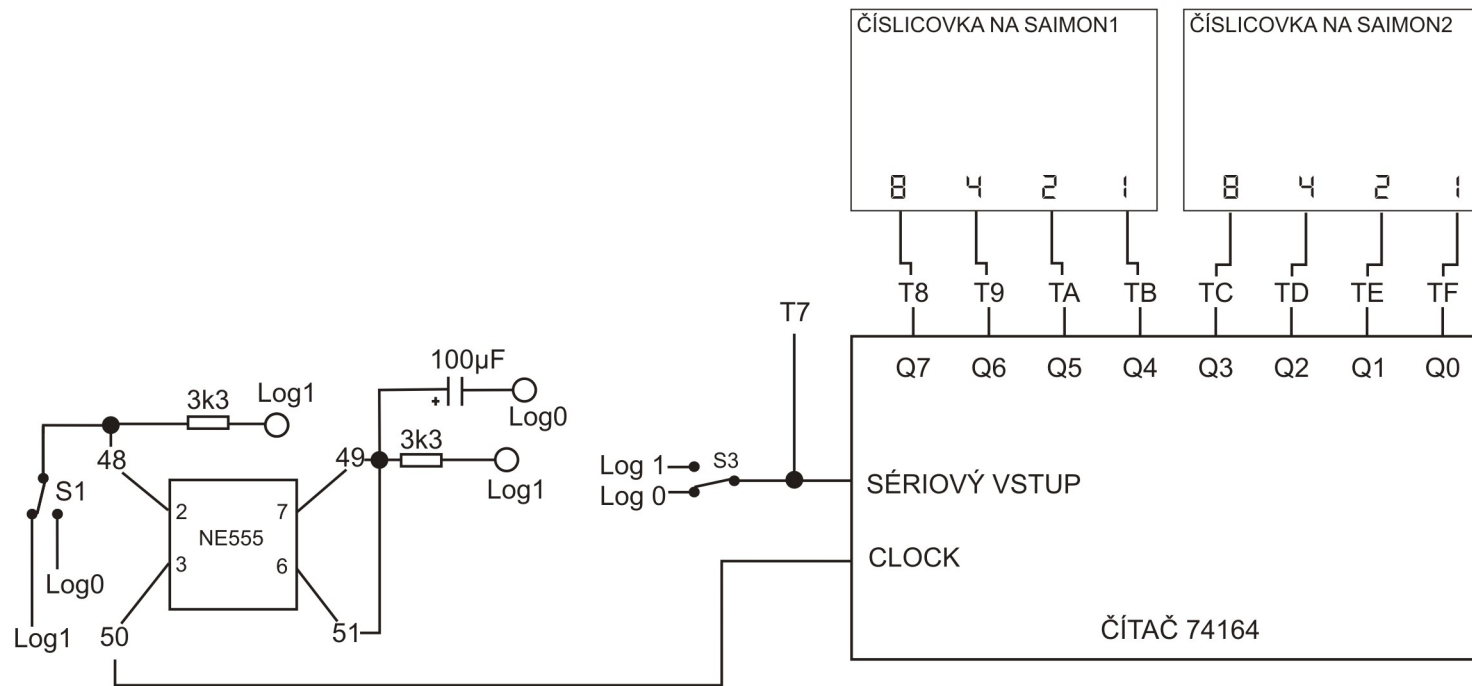


Obrázek 3.1 – Posuvný registr s výstupy na testovacích ledkách

Pojďme nastavit paralelní výstupy. Budeme chtít zobrazit paralelní kombinaci 10110010. Spínačem S3 nastavíme hodnotu prvního bitu na sériovém vstupu. Počítá se zleva, takže to bude hodnota log1. Aby čítač poznal, že je to první bit, musíme mu to sdělit. A to provedeme hodinovým pulzem přivedeným na zdířku CLOCK. V tu chvíli se hodnota bitu na sériovém vstupu objeví na výstupu Q0. Dále nastavíme hodnotu druhého bitu na sériovém vstupu, v tomto případě log0. A vygenerujeme hodinový signál. Obvod 74164 posune hodnotu všech svých výstupů doleva, čili co dříve bylo na výstupu Q0 teď bude na výstupu Q1. Na výstup Q0 nastaví hodnotu na sériovém vstupu, tedy log0. A takto budeme postupovat dále. Poznámka – posuvný registr tedy posouvá všechny své výstupy doleva, čili i to co dříve bylo například na výstupu Q5 bude po přijetí hodinového pulsu na výstupu Q6 atd.



Na obrázku 3.2 je posuvný registr, jehož výstupy jsou zároveň připojené na číslicovky.



Obrázek 3.1 – Posuvný registr s výstupy na testovacích ledkách

Čtyři bity zleva přivedeme na číslicovku na Saimon 1 a druhé čtyři bity vpravo přivedeme na číslicovku na Saimon 2. Úkolem bude zobrazit číslo 210. Jeho binární zápis je 11010010. Číslicovka zobrazuje hexadecimální tvar čísla. Bity 1101 znamenají hexadecimální znak „D“ a bity 0010 znamenají hexadecimální znak „2“. Postupně posílejte bity zleva na sériový vstup. Pokud uděláte vše správně, musí se D2 zobrazit na číslicovkách.

## **Kapitola 4 – Dvouřádkový displej s řadičem HD44780**

Protože s modulem Saimon 3 se věnujeme rozdílu mezi paralelním a sériovým přenosem informace, přidal jsem na něj i dvouřádkový displej, který je primárně určený pro arduino. Arduino je programovatelná platforma, které se věnuje celý modul Saimon 4 a díky které dostala celá stavebnice úplně nový rozměr. Pomocí několika řádkového programu pro arduino dokážeme zobrazit na displeji jakýkoliv text během chvíličky a jsou i tací, kteří pomocí tohoto displeje naprogramovali střílečku (fakt, nekecám, viděl jsem to).

Důležité je, že do tohoto displeje se posílá informace paralelně. A to osm bitů zároveň, nebo čtyři bity zároveň. Ta druhá možnost je náš případ. Saimon 4 obsahuje kromě arduina také displej, do kterého se posílá informace sériově.

Možná už je to nyní jasné, jaký je jeden z rozdílů paralelního a sériového přenosu, ale pro jistotu to zdůrazním – pro sériový přenos informace potřebujeme podstatně méně drátků :-).

V tomto dílu návodu vám ukážu jen základy psaní na displeji pomocí posílání paralelní informace s použitím spínačů. Celý návod naleznete na adrese zde (adresa platná v době psaní návodu)

<http://www.elektronika.kvalitne.cz/ATMEL/necoteorie/LCDmatice.html>

Položme si otázku - co je našim cílem, pokud máme displej? Napsat text, že ano, tedy zobrazit znaky. Takže začneme. Displej je potřeba zapnout a inicializovat. K tomu musíme umět dát displeji příkaz. Informace, kterou budeme displeji posílat, bude tedy muset být dvojího druhu. Buď mu budeme posílat příkazy, nebo kódy znaků, které má zobrazit. A ještě chci upozornit, že já se budu věnovat pouze čtyřbitové komunikaci s displejem.

Displej má na stavebnici následující vývody:

VCC – napájení, připojujeme na +5V

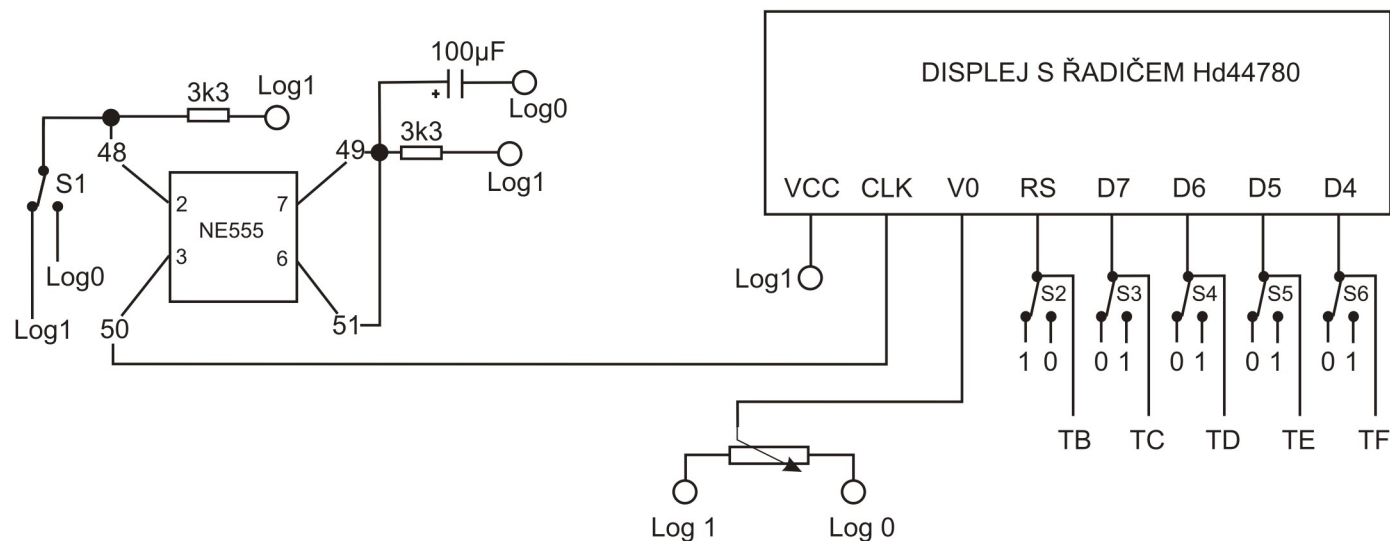
CLK – vstup pro hodinové pulzy, připojujeme obvykle na výstup 555

V0 – nastavení kontrastu displeje, připojujeme na jezdcu potenciometru

RS – zde říkáme displeji, zda chceme poslat příkaz (nastavíme na log 0) nebo kód znaku, který chceme zobrazit (nastavíme log 1)

D7, D6, D5, D4 – datové zdířky pro informaci

Zapojme obvod podle schéma na obrázku 4.1.



Obrázek 4.1 – Zapojení paralelního displeje

Jako zdroj hodinových pulzů použijeme opět monostabilní zapojení časovače NE555. Spínačem S1 vygenerujeme hodinový pulz. Potenciometrem nastavíme úroveň kontrastu tak, aby byl displej dobře čitelný. A jdeme na posílání příkazů.

Pro posílání příkazu musíme držet stisknutý spínač S2 (Poznámka – ve schématu můžete vidět, že stisknutý spínač S2 nastavuje logickou 0, zatímco barevné spínače, pokud jsou stisknuté, jsou zapojené tak, aby nastavovaly logickou 1). Postupujte po následujících krocích.

1. Displej inicializujeme. Stiskněte a držte S2. Datové spínače (to jsou S3,S4,S5,S6) nastavte na 0010 a vygenerujte pulz stiskem S1. To samé musíte na displej zaslat ještě jednou, takže znovu stiskněte S1. Datové spínače přenastavte na 1000. Stiskněte S1. Displej se nastaví na dvouřádkový a na čtyřbitovou komunikaci.
2. Displej smažeme, nastavíme kurzor na začátek. Stále držte S2. Datové spínače nastavte na 0000. Stiskněte S1. Dále nastavte 0001. Stiskněte S1.
3. Zapneme kurzor a nastavíme blikání kurzoru, abychom ho viděli. Datové spínače nastavte na 0000. Stiskněte S1. Dále nastavte 1111. Stiskněte S1. Nyní můžeme posílat kódy znaků. S2 už můžete pustit.

Všimněte si, že vždycky posíláme dvě čtveřice dat. Stejně to bude i pro znaky. Po každé čtveřici stiskněte S1. Naším úkolem bude napsat text, bez kterého si to snad nikdo ani nedokáže představit, a to bude „Ahoj svete!“.

Pro zaslání kódů znaků stačí pustit S2, datové spínače nastavit na první kombinaci, generovat pulz, nastavit datové spínače na druhou kombinaci a generovat druhý pulz. Zde jsou kódy znaků. Kdo by chtěl napsat jiný text, tak na konci uvedu tabulku, která je zkopírovaná z výše uvedeného odkazu.

Kódy pro náš text:

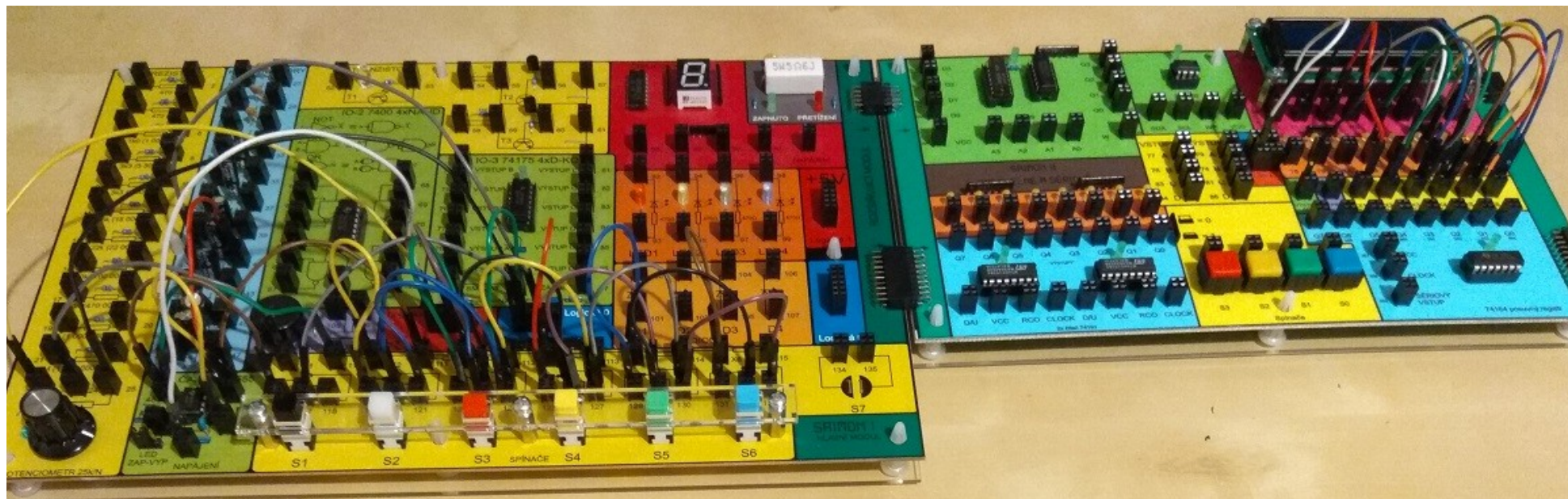
A = 0100 0001  
h = 0110 1000  
o = 0110 1111  
j = 0110 1010  
mezera = 0010 0000  
s = 0111 0011  
v = 0111 0110  
e = 0110 0101  
t = 0111 0100  
e = 0110 0101  
! = 0010 0001

Pokud uděláte chybu, je potřeba posunout kurzor zpět. To uděláte tak, že stiskem S2 přejdete do příkazového režimu a zašlete data 0001 0000 (posuneme kurzor vlevo). Pokud by někdo chtěl posunout kurzor vpravo, stačí zaslat 0001 0100. Celý displej smažeme už známým příkazem 0000 0001.

	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
xxxx000			0	@P`P							-	9	3	α	p	
xxxx001			!	1RQa9							.	7	4	ä	q	
xxxx010			"	2BRbr							「	イ	ツ	×	β	θ
xxxx011			#	3CScs							」	ウ	フ	ε	ε	*
xxxx100			\$	4DTdt							、	エ	ト	ト	μ	Ω
xxxx101			%	5EUeu							.	オ	ナ	1	σ	ü
xxxx110			&	6FUfv							ヲ	カ	ニ	ヨ	ρ	Σ
xxxx111			'	7GWgw							ヲ	キ	ヲ	ラ	g	π
xxxx1000			(	8HXhx							イ	ク	ネ	リ	」	×
xxxx1001			)	9IYiy							ウ	ケ	ノ	ル	'	4
xxxx1010			*	:JZjz							エ	コ	ハ	レ	j	≠
xxxx1011			+	;K[k{							*	サ	ヒ	ロ	*	≠
xxxx1100			,	<L[l							ハ	シ	フ	ワ	≠	≠
xxxx1101			-	=M]m}							ユ	ズ	ハ	シ	≠	÷
xxxx1110			.	>N^n→							ヨ	セ	ホ	°	ñ	
xxxx1111			/	?0_o←							ウ	ツ	マ	°	ö	■

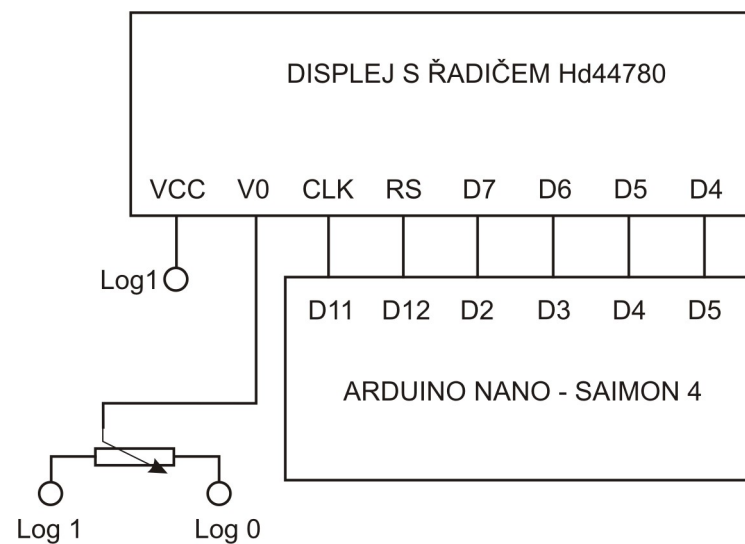
Obrázek 4.2 – Tabulka znaků displeje

Dále ještě fotka, jak vypadá zapojení na stavebnici podle schématu z obrázku 4.1. Je vidět, že pro všechny spínače, potenciometr i časovač jsem použil propojky.



Obrázek 4.3 – Fotka zapojení

Dále pro veliké zvědavce a majitele modulu Saimon 4 uvádím schéma zapojení displeje s arduinem a program, který text zobrazí.



Obrázek 4.4 – Zapojení displeje s arduinem

Program je upravená verze vzorového programu přímo z webových stránek arduino.cc <https://www.arduino.cc/en/Tutorial/LiquidCrystalDisplay>

```
#include <LiquidCrystal.h>
const int rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);

void setup() {
  lcd.begin(16, 2);
  lcd.print("Ahoj svete!");
}
void loop() {
}
```

## Kapitola 5 – Sériová paměť 24LC02

Tento integrovaný obvod je permanentní paměť. To, co do ní zapíšeme, v ní zůstane, dokud to nesmažeme nebo nepřepíšeme. Její ovládání pomocí spínačů mi nakonec přišlo příliš složité (v době plánování modulu Saimon 3 jsem to ještě nevěděl), tak uvádím pouze ovládání pomocí modulu arduino. To, co chci ukázat, je to, že veškeré ovládání se děje pouze pomocí dvou drátků. Pro zapojení ani nebudu uvádět schéma. Zapojte VCC na log1, WP na log0, SCL na A5 a SDA na A4. Pomocí následujícího programu můžete zapisovat a číst data.

```
#include <Wire.h>
//funkce pro zápis
void i2c_eeprom_write_byte( int deviceaddress,int eeaddress, byte data ) {
    int rdata = data;
    Wire.beginTransmission(deviceaddress);
    Wire.write(eeaddress); // odešle vyšší byte
    Wire.write(rdata);
    Wire.endTransmission();
}

//funkce pro čtení
byte i2c_eeprom_read_byte( int deviceaddress, int eeaddress ) {
    byte rdata = 0xFF;
    Wire.beginTransmission(deviceaddress);
    Wire.write(eeaddress); // odešle vyšší byte
    Wire.endTransmission();
    Wire.requestFrom(deviceaddress,1);
    if (Wire.available()) rdata = Wire.read();
    return rdata;
}
```



```
void setup()
{
  Wire.begin();
  Serial.begin(9600);

  i2c_eeprom_write_byte(0x50, 0, B10110010);
  delay(5);
  byte a = i2c_eeprom_read_byte(0x50, 0);
  Serial.print("Hodnota je: ");
  Serial.println(a);
}

void loop() {}
```

## Obsah

Úvod.....	2
Kapitola 1 – Paralelní paměť 16x4 bity.....	3
Kapitola 2 – Čítač 74191.....	6
Kapitola 3 – Posuvný registr 74164.....	8
Kapitola 4 – Dvouřádkový displej s řadičem HD44780.....	10
Kapitola 5 – Sériová paměť 24LC02.....	15