

Parallel to Serial Shifting IN

Com o circuito integrado CD4021B

Objectivo:

Quando num projecto o número de sensores digitais é superior ao número de pinos de entradas digitais disponíveis no microcontrolador (também designado MCU), uma técnica utilizada é recorrer a circuitos integrados "parallel to serial shift register" de 8 ou 16 bits.

Assim, por cada entrada digital do MCU, podemos controlar 8 ou 16 sensores digitais, em vez de um só (Shifting In).

Realizando a montagem em cascata, estes valores poderão passar para 16, 24, etc., ou 32, 48, etc., respectivamente, à custa de mais complexidade no circuito electrónico e da programação do MCU.

A questão inversa também é possível (Shifting Out), isto é, com um só pino digital do MCU de output e um circuito integrado de 8 ou 16 bits, controlar, por exemplo, o funcionamento de uma matriz de led's.

Equipamentos/componentes electrónicos:

- 1 Microcontrolador Arduino Mega 2560, UnoRev3 ou compatível
- 1 Circuito integrado CD4021B
- 8 Interruptores de botão
- 8 Resistências de 10 k 1/4W
- 1 Speaker

Microcontrolador Arduino Mega 2560



- Microcontroller ATmega2560
- Operating Voltage 5V
- Input Voltage (recommended) 7-12V
- Input Voltage (limits) 6-20V
- Digital I/O Pins 54 (of which 15 provide PWM output)
- Analog Input Pins 16
- DC Current per I/O Pin 40 mA
- DC Current for 3.3V Pin 50 mA
- Flash Memory 256 KB of which 8 KB used by bootloader
- SRAM 8 KB
- EEPROM
- 4 KB (which can be read and written with the EEPROM library)
- Clock Speed 16 MHz
- Quando alimentado com um transformador ligado a 220 Volts, devem ser rigorosamente observados as seguintes regras:
 - Output: de 7 a 12 Volts
 - Jack de 2.1 mm *com o centro positivo*

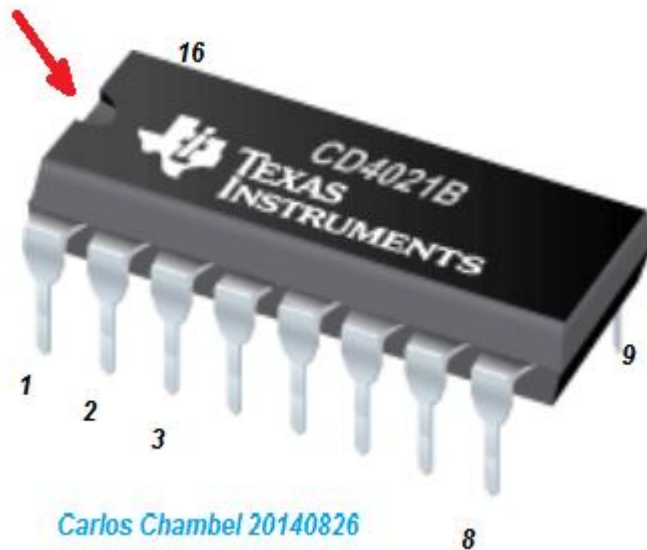
Esquema: [arduino-mega2560_R3-schematic](#)

Diagrama dos pinos: [PinMap2560 page](#)

Circuito integrado **CD4021B**

CMOS 8-Stage Static Shift Register

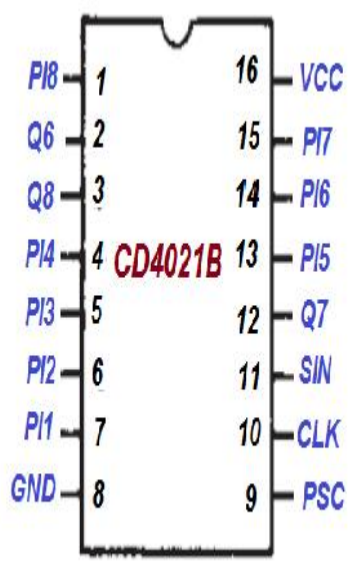
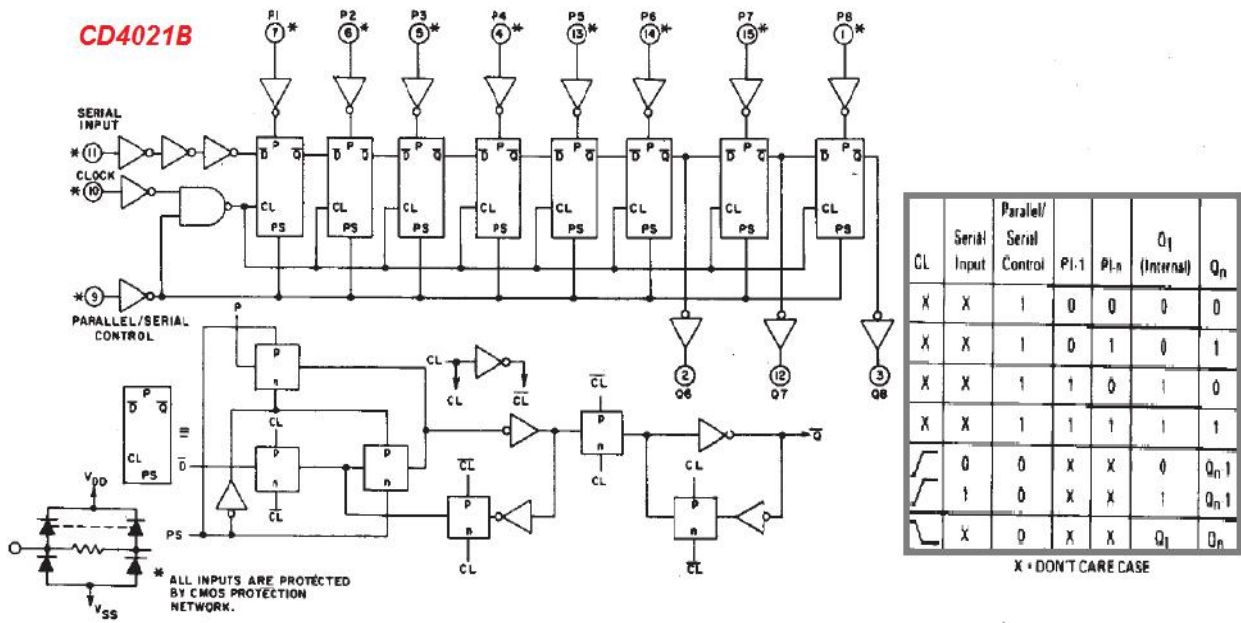
Trata-se de um circuito integrado com 8 bits (1 byte) de memória



Características técnicas:

- Technology Family: CD4000
- VCC (Min): 3V
- VCC (Max): 18V
- Voltage (Nom): 10V
- F @ Nom Voltage (Max): 8Mhz
- ICC @ Nom Voltage (Max): 0.3mA
- tpd @ Nom Voltage (Max): 160ns
- Output Drive (IOL/IOH) (Max): 0.5/-1.5 mA
- Input Type: CMOS
- Output Type: CMOS
- Operating Temperature Range: -55° TO 125° C
- Pin/Package: 16 PDIP

Diagrama lógico e tabela



Carlos Chambel

<i>PINOS 7,6,5,4,13,14,15,1</i>	PI1, PI8	Parallel Inputs
<i>PINO 8</i>	GND	Ground
<i>PINO 9</i>	PSC	Parallel/Serial Control (latch pin)
<i>PINO 10</i>	CKL	Shift register clock
<i>PINO 11</i>	SIN	Serial data input
<i>PINO 16</i>	VCC	DC supply voltage
<i>PINOS 2, 12, 3</i>	Q6, Q7, Q8	Serial Output

Como funciona:

O circuito integrado está ligado ao MCU por três pinos:

- O pino número **3** (Q8) que liga ao pino digital **9** do microcontrolador. Este pino número 9 tem que ser definido para input de dados. - **Data Pin**
- O pino número **9** (P/S C - Parallel to Serial Control) que liga ao pino digital **8** do microcontrolador. Este pino número 8 tem que ser definido para output e serve para controlar quer o preenchimento dos registos com os seus inputs (PINOS PI1 a PI8), quer a transferência da informação. - **Latch Pin**
- O pino número **10** (CLOCK) que liga ao Pino Digital **7** do microcontrolador. Este pino número 7 tem que ser definido para output e serve de relógio para controlar a comunicação entre os dois dispositivos. - **Clock Pin**

```

// Carlos Chambel
// 2014 09 03
// versão 1.3
//
#include "pitches.h"
int latchPin = 8;
int dataPin = 9;
int clockPin = 7;
int speakerPin = 5;
int Notasmu[] = {
    NOTE_B6, NOTE_C7,NOTE_CS7,NOTE_D7,NOTE_DS7, NOTE_E7, NOTE_F7,NOTE_FS7};
int Notasdu = 4 ;
int Notasdus[] = {
    4, 8, 8, 4,4,4,4,4 };
byte switchVar1 = 72; //01001000
int thisNote = 0;
void setup() {
    Serial.begin(9600);
    pinMode(latchPin, OUTPUT);
    pinMode(clockPin, OUTPUT);
    pinMode(dataPin, INPUT);
}

```



Existem três etapas em todo o processo:

- *O circuito integrado trabalha de uma forma assíncrona e paralela quando está a receber os sinais digitais dos sensores através dos pinos PI1 a PI8, isto é:*
 - *não depende do microcontrolador para realizar esta função e,*
 - *de uma só vez recebe os sinais digitais dos sensores coleccionando os oito bits em cada posição da memória.*

Esta fase verifica-se quando se altera o valor do pino 8 (latchPin) para HIGH:

```

void loop()
{
  digitalWrite(latchPin,1);
  delayMicroseconds(20);
  digitalWrite(latchPin,0);
  switchvl = shiftIn(dataPin, clockPin);
  thisNote = switchvl;
  switch (thisNote)
  {
  case 1:
    thisNote = 1;
    Serial.print("byte ...");
    Serial.print(thisNote);
    break;
  }
}

```




- Quando, passados vinte microsegundos, se muda o valor do pino 8 (latchPin) para LOW estão reunidas as condições para a transferência da informação:

```

void loop()
{
  digitalWrite(latchPin,1);
  delayMicroseconds(20);
  digitalWrite(latchPin,0);
  switchvl = shiftIn(dataPin, clockPin);
  thisNote = switchvl;
  switch (thisNote)
  {
  case 1:
    thisNote = 1;
    Serial.print("byte ");
  }
}

```



- Nesta fase, o circuito integrado funciona de uma forma síncrona e serial, isto é:
 - síncrona, porque o circuito integrado funciona de uma forma sincronizada com o relógio gerado pelo microcontrolador e,
 - serial porque a transferência é efectuada bit a bit por cada ciclo.

O microcontrolador envia ao circuito integrado, através do pino 9, a cada 0,2 microsegundos, um sinal LOW ou HIGH, e cada vez que isto acontece o circuito integrado avança uma posição no registo. A informação vai sendo lida e colecionada na variável kDataIn bit a bit:

```

for (i=7; i>=0; i--)
{
  digitalWrite(kClockPin, 0);
  delayMicroseconds(0.2);
  tempt = digitalRead(kDataPin);
  if (tempt) {
    pinState = 1;
    kDataIn = kDataIn | (1 << i);
  }
  else {
    pinState = 0;
  }
  digitalWrite(kClockPin, 1);
}

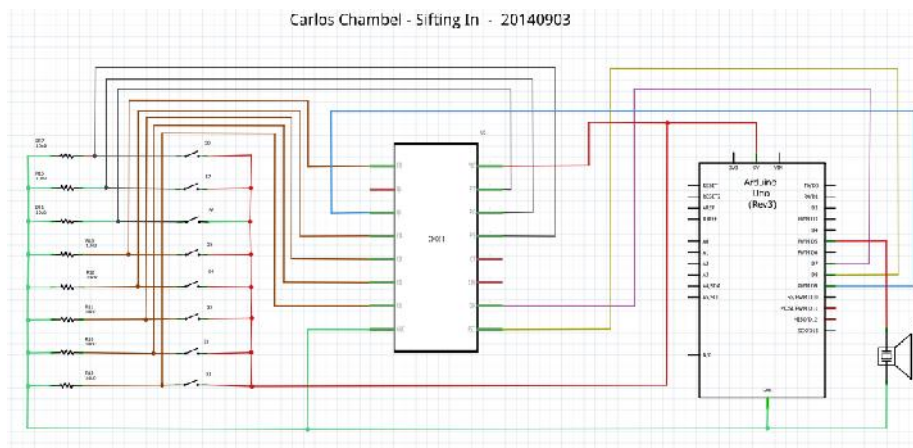
```

Montagem:

- Conexões do circuito integrado CD4021B ao microcontrolador Arduino Mega 2560, Uno Rev3 ou compatível:
 - Pino 8 (gnd) ligar ao Pino gnd
 - Pino 16 (vcc) ligar ao Pino 5V
 - Pino 3 (Q8) ligar ao Pino Digital 9 - **Data Pin**
 - Pino 9 (P/S C) ligar ao Pino Digital 8 - **Latch Pin**
 - Pino 10 (CLOCK) ligar ao Pino Digital 7 - **Clock Pin**
 - *Pinos 7,6,5,4,13,14,15,1* PI1, PI8 ligar aos interruptores e resistências - **Parallel Inputs**
 - Positivo do speaker ao Pino Digital 5
 - Negativo do speaker a GND (0V)
 - Ligar os positivos dos interruptores a +5V

Esquema desenhado para este projecto:

(Para melhor consulta, aceder ao respectivo link)



Testes:

Foram efectuados os testes e enviados para um PC, através de uma porta COM, os seguintes dados:

- Bit em notação binária correspondente ao interruptor activado,
- Bit em notação decimal,
- Valor da nota musical associada a esse bit.

Filme no YouTube:

- Podem ser visualizados os valores dos testes acima e o funcionamento do projecto (aceder ao respectivo link).