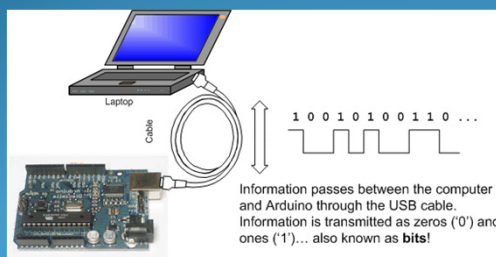


UART
I2C

Comunicação Serial com Arduino

Prof. Cláudio A. Fleury

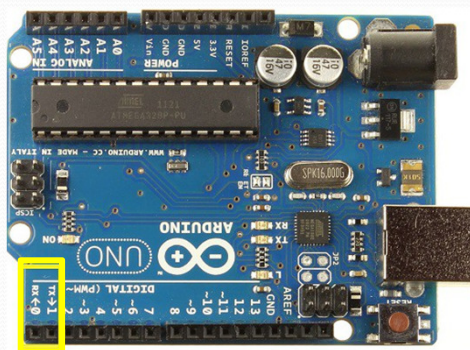
Nov-2011



55 Slides

Pinos para Comunicação Serial

- Os sinais usados na comunicação Serial do Arduino UNO são do tipo TTL (0 a 5V) e gerados pela UART* do microcontrolador
- Para se comunicar serialmente com um PC, ou outro dispositivo, que não tenha o mesmo nível de tensão é necessário usar um módulo conversor: TTL/ RS232, TTL/RS485, TTL/USB, TTL/3,3V...

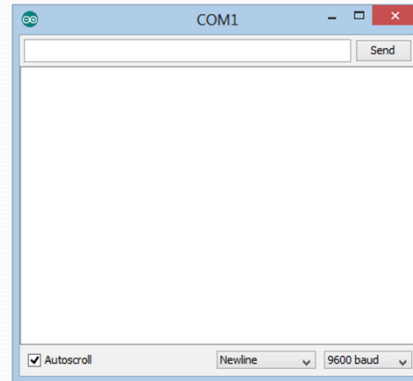


Pino 0 → RX
Pino 1 → TX

* Universal Asynchronous Receiver/Transmitter

Terminal/Monitor Serial

- Porta Serial UART do Arduino (pinos 0 e 1)
- Usada pela IDE no *upload* de *sketches* → gravação dos programas na memória Flash
- IDE oferece um **Terminal Serial** que auxilia na recepção e envio de dados para a placa sem a necessidade de se recorrer a uma ferramenta externa



Prof. Cláudio A. Fleury - Nov-2011

3

Peer-to-Peer = P2P

Comunicação Ponto a Ponto

Comandos

- Configuração da Porta Serial

```
// configura a porta serial do Arduino
Serial.begin(9600, SERIAL8N1);
```

- Quantidade de bytes disponíveis para leitura na Porta Serial

```
// bytes disponíveis para leitura
n = Serial.available();
```

- Função executada quando existir dados disponíveis

```
// executada qdo houver dados disponíveis
void serialEvent() {
    // comandos
}
```

default →

SERIAL_5N1
SERIAL_6N1
SERIAL_7N1
SERIAL_8N1
SERIAL_5N2
SERIAL_6N2
SERIAL_7N2
SERIAL_8N2
SERIAL_5E1
SERIAL_6E1
SERIAL_7E1
SERIAL_8E1
SERIAL_5E2
SERIAL_6E2
SERIAL_7E2
SERIAL_8E2
SERIAL_5O1
SERIAL_6O1
SERIAL_7O1
SERIAL_8O1
SERIAL_5O2
SERIAL_6O2
SERIAL_7O2
SERIAL_8O2

Prof. Cláudio A. Fleury - Nov-2011

4

Comunicação Ponto a Ponto

Peer-to-Peer = P2P

Comandos

- Configuração da Porta Serial

```
// configura a porta serial do Arduino
Serial.begin(9600);
```

- Qtde de bytes disponíveis para leitura na Porta Serial

```
// bytes disponíveis para leitura
n = Serial.available();
```

Prof. Cláudio A. Fleury - Nov-2011

5

Comandos de Entrada de Dados

- Leitura de um byte na entrada da Porta Serial
(remove o primeiro byte do buffer de entrada da porta serial)

```
// lê um byte da porta serial
carac = Serial.read();
// carac == -1 se não existir byte a ser lido
```

- Leitura de um byte na entrada da Porta Serial
(não remove o byte do buffer de entrada da porta serial)

```
// retorna o primeiro byte do buffer serial
carac = Serial.peek();
// chamadas sucessivas a essa função retorna
// o mesmo byte, o primeiro do buffer
```

Prof. Cláudio A. Fleury - Nov-2011

6

Comandos de Saída de Dados

- Gravação de um byte “binário” na saída da Porta Serial

```
// escrita de um byte binário na porta serial
Serial.write(byte);
int bytesEnv = Serial.write("Arduino");
```

- Gravação de um byte “ASCII” na saída da Porta Serial

```
// escrita de um byte ASCII na porta serial
Serial.print(carac);
// idem, porém com <CR/LF> ao final da escrita
Serial.println(string);
```

```
Serial.print(78)           // envia "78"
Serial.print(1.23456)     // envia "1.23"
Serial.print('N')         // envia 'N'
Serial.print("Olá!")      // envia "Olá!"
```

Prof. Cláudio A. Fleury - Nov-2011

7

Exemplo

```
// Timer Regressivo Fixo (5'15") com exibição na porta serial
// Prof. Cláudio - Out/2014
long tempo;
int minu = 5, segu = 15;

void setup() {
  Serial.begin(9600);
  tempo = millis();
}

void loop() {
  if(millis()-tempo > 100) {
    tempo = millis();
    Serial.print("Timer: ");
    Serial.print(minu); Serial.print(":"); Serial.println(segu);
    atualizaTempo();
  }
}

void atualizaTempo(void) {
  if(--segu < 0) {
    segu = 59;
    if(--minu < 0) {
      Serial.println("Tempo esgotado!");
      while(1);
    }
  }
}
```

Prof. Cláudio A. Fleury - Nov-2011

8

Exercício 1

- Considerando o Timer Regressivo do exemplo anterior, o usuário deverá informar o intervalo de temporização desejado (minutos e segundos, dois bytes cada) no monitor serial do Arduino, antes de iniciar a contagem regressiva de tempo.

dica: use os comandos

`Serial.read()`

`Serial.available()`

```
// Timer Regressivo com exibição na porta serial
// Prof. Cláudio - Set/2016

// variáveis globais -----
long tempo;
int minu = 1, segu = 5;

void setup() {
  Serial.begin(9600); tempo = millis();
  programaTempo();
}

void loop() {
  if(millis()-tempo > 100) {
    tempo = millis();
    Serial.print("Timer: "); mostra(minu); Serial.print(":");
    mostra(segu); Serial.println(""); atualizaTempo();
  }
}

// funções auxiliares -----
void atualizaTempo(void) {
  if(--segu < 0) {
    segu = 59;
    if(--minu < 0) {
      Serial.println("Tempo esgotado!");
      while(1); } } }

void mostra(int x) { // mostra valor 'x' com duas casas
  if (x < 10)
    Serial.print("0");
  Serial.print(x);
}

void programaTempo(void) { // aguarda envio do tempo (mm,ss) pela Serial
  int dzn, und;
  Serial.print("Minutos (mm):"); while(Serial.available() < 2);
  dzn = Serial.read()-'0'; und = Serial.read()-'0'; minu = 10*dzn + und;
  Serial.print("Segundos (ss):"); while(Serial.available() < 2);
  dzn = Serial.read()-'0'; und = Serial.read()-'0'; segu = 10*dzn + und;
  Serial.print(minu); Serial.print(":"); Serial.println(segu);
}
```

9

Exercício 2

- Acrescentar ao Timer do exercício anterior as seguintes funções:
 - Pausa ('P') – interrompe a temporização momentaneamente, até que o comando 'C' seja recebido pela porta serial.
 - Continua ('C') – continua a temporização a partir do ponto em que havia sido interrompida pelo comando 'P'.
 - Reprogramação ('R') – reprograma o tempo, reiniciando a temporização.

```
// Timer Regressivo Fixo com exibição na porta serial
// Prof. Cláudio - Set/2016
long tempo; int minu, segu;

void setup() {
  Serial.begin(9600); programaTempo(); tempo = millis(); }

void loop() {
  if(millis()-tempo > 100) {
    tempo = millis();
    Serial.print("Timer: "); mostra2C(minu);
    Serial.print(":"); mostra2C(segu); Serial.println("");
    atualizaTempo(); }
  if(Serial.available() > 0)
    switch(Serial.read()) {
      case 'P': while(Serial.read() != 'C'); // 'P' pausa o temporizador
                break; // 'C' continua
      case 'R': programaTempo(); // 'R' reprograma o temporizador
    }
}

void atualizaTempo(void) {
  if(--segu < 0) {
    segu = 59;
    if(--minu < 0) {
      Serial.println("Tempo esgotado!");
      while(1); } } }

void mostra2C(int x) { // mostra valor 'x' com duas casas
  if (x < 10) Serial.print("0");
  Serial.print(x);
}

int leiaNumSerial(int digitos) {
  int dzn, und;
  while(Serial.available() < 2);
  dzn = Serial.read()-'0'; und = Serial.read()-'0';
  return(10*dzn + und);
}

void programaTempo(void) {
  Serial.println(); Serial.println(">>> Temporizador <<<");
  Serial.print("Minutos (99): "); minu = leiaNumSerial(2); Serial.println(minu);
  Serial.print("Segundos (99): "); segu = leiaNumSerial(2); Serial.println(segu);
}
```

Prática 0

Depuração de *Sketchs*
Envio de Informações do Arduino para o PC

Prof. Cláudio A. Fleury - Nov-2011

11

Prática 0

- Mostra mensagens e/ou valores de variáveis no Monitor Serial do IDE do Arduino

```
/*
 * Saída Serial: envia valores numéricos p/ porta serial
 */

void setup() {
  Serial.begin(9600);      // envia/recebe a 9600 baud
}

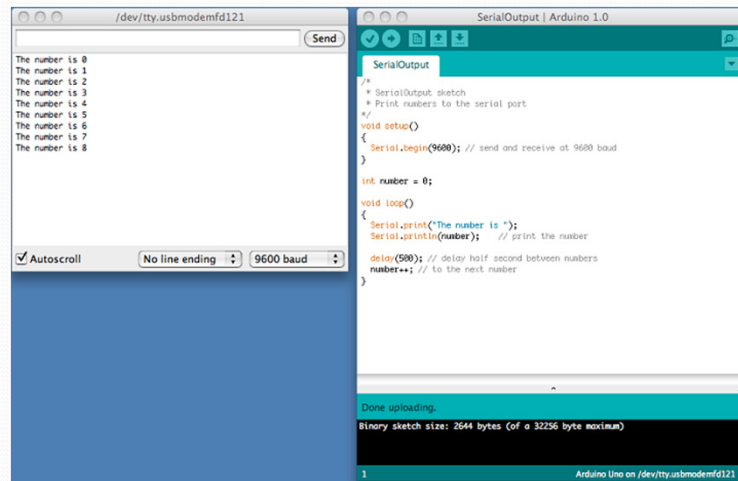
int num = 0;

void loop() {
  Serial.print("Número: ");
  Serial.println(num);     // mostra o número no monitor
  delay(500);              // atraso meio seg. entre números
  num++;                  // próximo número
}
```

Prof. Cláudio A. Fleury - Nov-2011

12

Prática 0



Experimente outro programa de acesso a porta serial como o COOLTERM.

Prof. Cláudio A. Fleury - Nov-2011

13

Prática 1

Uso da função `SerialEvent()`

Prof. Cláudio A. Fleury - Nov-2011

14

Prática 1

- `SerialEvent()` é chamada depois de uma passada do `loop()` se existir algum byte disponível no *buffer* de entrada da Porta Serial
- Quando novos dados seriais chegam, o *sketch* adiciona-os numa String
- Quando um carácter ASCII 'linhanova' (LF – linefeed – '\n') é recebido, o laço envia o conteúdo da *string* para a saída da porta serial, e limpa a variável para iniciar um novo ciclo
- Um bom teste desse *sketch* seria usá-lo com um receptor GPS que envia sentenças NMEA 0183

Prof. Cláudio A. Fleury - Nov-2011

15

Prática 1

```

/*
 * Comunicação Serial: usando a função SerialEvent()
 */
String stringEntr = "";           // string p/ dados de entr.
boolean stringCompleta = false;   // se a string está completa

void setup() {
  Serial.begin(9600);             // envia/recebe a 9600 baud
  stringEntr.reserve(200);        // reserva 200 bytes para string
}

void loop() {
  // envia a string quando chega um carácter 'newline':
  if (stringCompleta) {
    Serial.println(stringEntr);
    stringEntr = "";              // limpa a var. string
    stringCompleta = false;
  }
}

```

Prof. Cláudio A. Fleury - Nov-2011

16

Prática 1

```

/*
SerialEvent ocorre sempre que chegam novos dados entrada
serial (pino RECEPTOR). Esta rotina é executada após cada
loop() → o uso de delay() no loop() pode atrasar a resposta.
Vários bytes de dados podem estar disponíveis.
*/

void serialEvent() {
  while (Serial.available()) {
    char inChar = (char)Serial.read();    // pega novo byte
    stringEntr += inChar;                // acrescenta-o à stringEntr
    // se o caracter lido for um newline, então liga um flag
    // de modo que o laço principal executará uma ação
    if (inChar == '\n')
      stringCompleta = true;
  }
}

```

Prof. Cláudio A. Fleury - Nov-2011

17

Prática 1

Use o monitor serial da IDE do Arduino para enviar dados ao Sketch...

The screenshot shows the Arduino IDE interface. The main window displays the code for 'EventoSerial' in Arduino 1.0.3. The code includes comments in Portuguese explaining the use of the SerialEvent function. The serial monitor on the right, titled 'COM28', shows the output of the sketch. The output text is as follows:

```

frase terminada com NEWLINE

Uso da função SerialEvent()

Arduino devolve os caracteres recebidos pela
porta serial quando encontra um caracter '\n'

```

The serial monitor also has an 'Autoscroll' checkbox checked at the bottom.

Prof. Cláudio A. Fleury - Nov-2011

18

Prática 2

- Comandar remotamente o acendimento proporcional de um LED ligado a uma porta de saída analógica

```
#include <LiquidCrystal.h>
// inicia LCD com os números dos pinos da interface
LiquidCrystal lcd(12,11,5,4,3,2); // RS, E, D4, D5, D6, D7
const int pinoLed = 9; // o pino no qual o LED está ligado
int pos = 0; // posição de armazenamento do caracter rxdo

void setup() {
  Serial.begin(9600); // inicia a comunicação serial
  pinMode(pinoLed, OUTPUT); // inicia o ledPin como saída
  pinMode(10, OUTPUT);
  lcd.begin(16, 2); // número de linhas e colunas do LCD: 16 x 2
  lcd.println("Aguardando CMD: "); lcd.print("999<ENTER>");
  Serial.println("Aguardando Comando Remoto (999<ENTER>): ");
}

int decodifica(char *s) { // int decodifica(char s[])
  int soma, i;
  for(soma=i=0; (s[i]!=0) && (i<pos); i++)
    soma = soma*10 + (s[i]-48);
  return soma;
}
```

Prof. Cláudio A. Fleury - Nov-2011

21

Prática 2

- Comandar remotamente o acendimento proporcional de um LED ligado a uma porta de saída analógica (continuação)

```
void loop() {
  char j, n, carac, seq[20];
  int brilho;

  n = Serial.available(); // qtde de dados enviados pelo remoto
  for(j=0; j<n; j++) {
    carac = Serial.read();
    if(carac == '.' || pos > 2) // lê bytes até encontrar CR (13 = 0x0D)
      break;
    seq[pos++] = carac;
  }
  if((carac == '.') || (pos > 2)) {
    seq[pos] = '\0';
    Serial.print(" - "); Serial.println(seq);
    brilho = decodifica(seq);
    constrain(brilho,0,255);
    pos = 0;
    lcd.setCursor(0, 1); lcd.print("          ");
    lcd.setCursor(0, 1); lcd.print(brilho);
    analogWrite(pinoLed, (byte)brilho); // ajusta o brilho do LED }
    while(Serial.available()) // algo para ser lido?
      char dados = Serial.read(); // se sim, leia-o
  }
}
```

Prof. Cláudio A. Fleury - Nov-2011

22

Prática 3

Jogo de Adivinhação

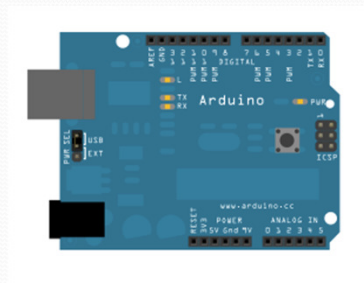
Prof. Cláudio A. Fleury - Nov-2011

23

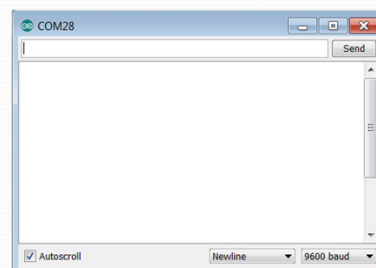
Prática 3

- Você escolhe um número entre 0 e 100 e o Arduino advinha o número...

ARDUINO



MONITOR SERIAL



Prof. Cláudio A. Fleury - Nov-2011

24

Prática 3

- Você escolhe um número entre 0 e 100 e o Arduino adivinhará...

```
/* Adivinha o número de 0 a 100 escolhido pelo usuário... */

boolean respondido = true;
int palpite, intervalo = 100;

void setup() {
  Serial.begin(9600);      // envia/recebe a 9600 baud
  Serial.println("Jogo de Adivinhacao de numero (0 a 100):\n");
  Serial.println("Responda as tentativas com '+' ou '-' se num. escolhido");
  Serial.println("por vc for > ou < que o indicado por mim (Arduino,");
  Serial.println("ou '.' (digito zero) se eu tiver adivinhado!");
  Serial.println();
  palpite = intervalo/2;
  intervalo /= 2;
}

void loop() {
  if(respondido) {
    Serial.print("ARDUINO: O numero eh ");
    Serial.print(palpite);
    Serial.print(" "? [+ . -]\n");
    respondido = false; }
}
```

Prof. Cláudio A. Fleury - Nov-2011

25

Prática 3

- Comandar remotamente o brilho de um LED... (continuação)

```
/* Adivinha número de 0 a 100 escolhido pelo usuário... */

void serialEvent() {
  while (Serial.available()) {
    char inChar = (char)Serial.read();    // pega resposta
    if (inChar == '+') {
      intervalo = max(1,intervalo/2);
      palpite += intervalo;
      respondido = true; }
    else if (inChar == '-') {
      intervalo = max(1,intervalo/2);
      palpite -= intervalo;
      respondido = true; }
    else if (inChar == '.') {
      Serial.print("\nARDUINO: Este eh o numero: ");
      Serial.println(palpite);
      Serial.println("ARDUINO: Vamos jogar denovo? (RESET-me)"); }
    else {
      Serial.print(inChar,HEX); Serial.println("???"); }
    delay(50);
    while (Serial.available()) {        // descarta bytes adicionais no buffer
      Serial.read(); } }
}
```

Prof. Cláudio A. Fleury - Nov-2011

26