

# Comunicação Serial com o AVR ATMEGA8

Vitor Amadeu Souza

vitor@cerne-tec.com.br

## Introdução

Os microcontroladores AVR vem a cada dia tomando cada vez mais espaço nos novos projetos eletrônicos microcontrolados. Isso de fato é em virtude do baixo custo do mesmo e da alta capacidade de processamento que este apresenta aliada de uma vasta documentação achada facilmente na internet. Neste artigo, irei apresentar ao leitor juntamente com o compilador BASCOM a facilidade e simplicidade em escrever um programa que possa transmitir e receber dados da porta serial, usando a placa AVR ATMEGA8 da Cerne Tecnologia e Treinamento ([WWW.cerne-tec.com.br](http://WWW.cerne-tec.com.br)).

## Características

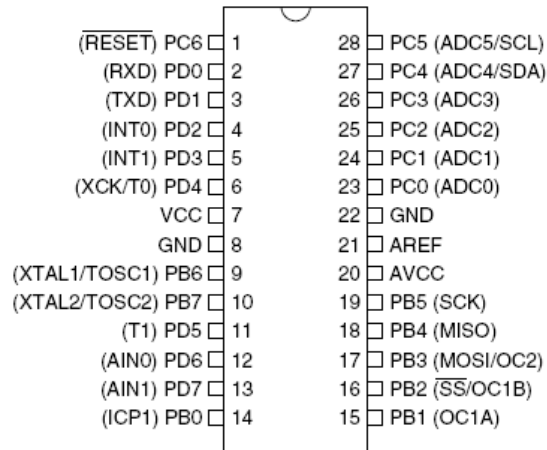
Vejamos de início os recursos disponíveis no microcontrolador AVR ATMEGA8:

- 130 Instruções de Programação Assembler;
- Processamento de até 16 MIPS (Milhões de Instruções por Segundo).
- 8Kbyte de memória de programa Flash que pode ser escrita até 10000 vezes;
- 1kB de memória de dados SRAM;
- 512 Bytes de memória EEPROM;
- 6 Entradas analógicas no modelo DIP;
- 23 Linhas de I/O Programáveis

Podemos ver pela lista acima que de fato este microcontrolador apresenta características bem atraentes que de fato poderão ser de grande ajuda em um projeto microcontrolado como por exemplo a memória de dados do mesmo que é de 1kByte. Estes microcontroladores também são facilmente achados no mercado, custando em média R\$ 10,00 na época em que este artigo foi escrito.

Podemos apreciar a pinagem do ATMEGA8 na figura abaixo:

## PDIP

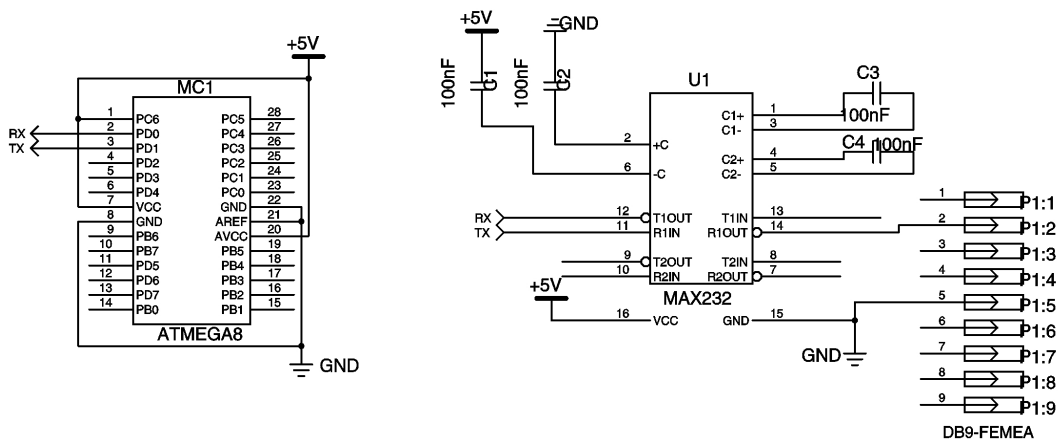


Observe pela pinagem apresentada, que o mesmo possui três PORTS (portas) de I/O identificadas por PORTB (Pinos 14, 15, 16, 17, 18, 19, 9 e 10), PORTC (Pinos 23, 24, 25, 26, 27, 28 e 1), PORTD (Pinos 2, 3, 4, 5, 6, 11, 12 e 13). Nestes PORTS podemos conectar elementos digitais como leds, displays, botões e etc assim como entradas analógicas que estão disponíveis no PORTC.

Este microcontrolador funciona com uma tensão de 5V porém a sua faixa de funcionamento é extensa, podendo variar de 2,7 a 5,5 V para o microcontrolador ATMEGA8L. No nosso experimento, iremos usar dois pinos em especial para a comunicação serial, que são os pinos 3 (TXD) e 2 (RXD).

## Esquema Elétrico

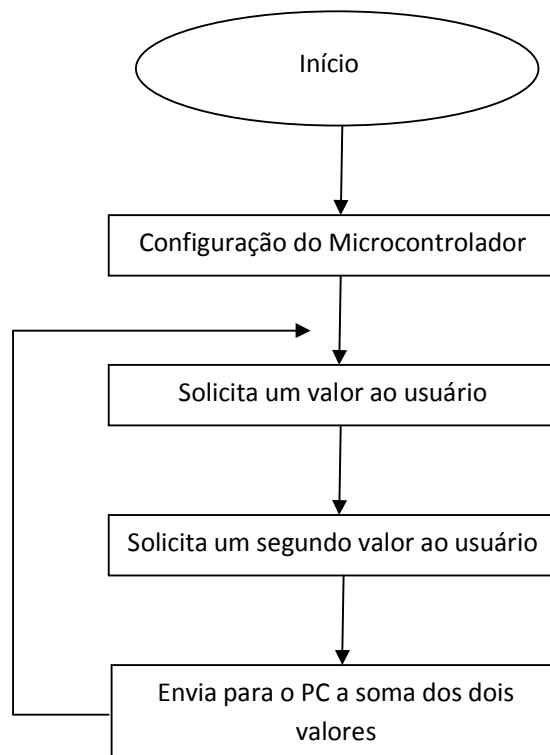
Vejamos abaixo o esquema elétrico que permitirá montar e testar o nosso projeto:



Note que o nosso esquema é bastante simples, tendo o microcontrolador ATMEGA8 assim como o circuito de conversão TTL – RS232 alimentados por 5V. O microcontrolador ATMEGA8 não precisa de clock já que o mesmo possui clock interno. Este esquema pode ser montado em uma pequena placa padrão ou até mesmo desenhado em uma placa de fenolite ou também testada na placa de desenvolvimento abordada no início do artigo.

### Fluxograma do Projeto

Abaixo está apresentado o fluxograma para funcionamento do projeto:

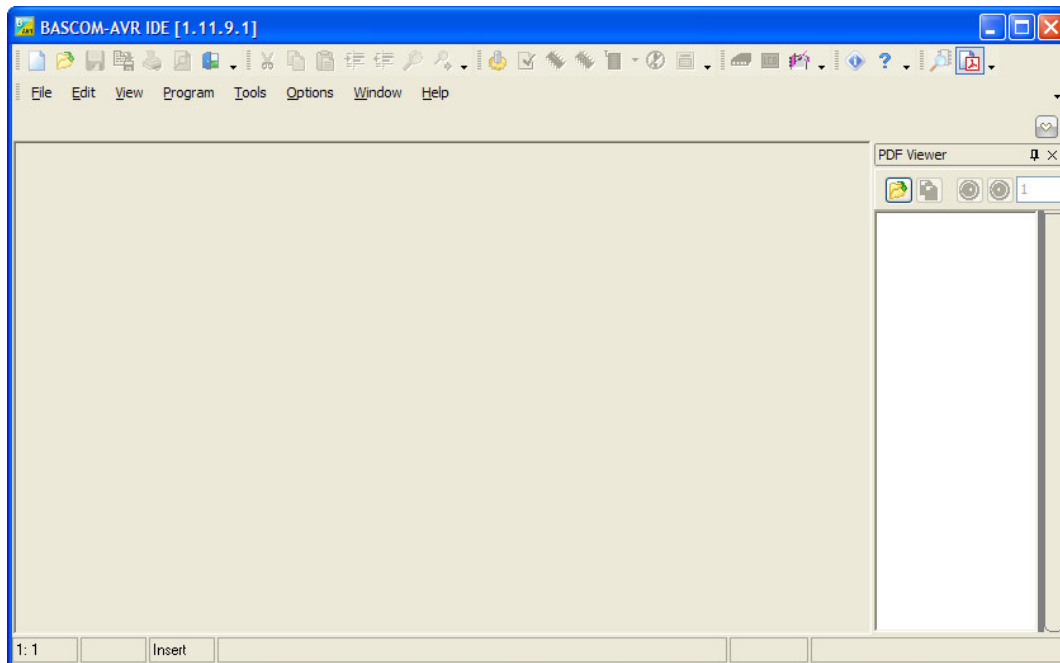


Veja que pelo fluxograma conseguimos ver com clareza como o projeto irá se comportar. Primeiramente, o microcontrolador irá solicitar um valor ao usuário no PC. Este irá informar algum valor seguido do pressionar da tecla enter. Em seguida, o programa irá solicitar um segundo valor. Após digitar o mesmo e pressionar Enter no PC, o microcontrolador irá somar os dois valores recebidos e retornar com o resultado para o PC via serial RS232, comprovando assim o envio e recebimento de dados via RS232 com o microcontrolador AVR e o compilador BASCOM.

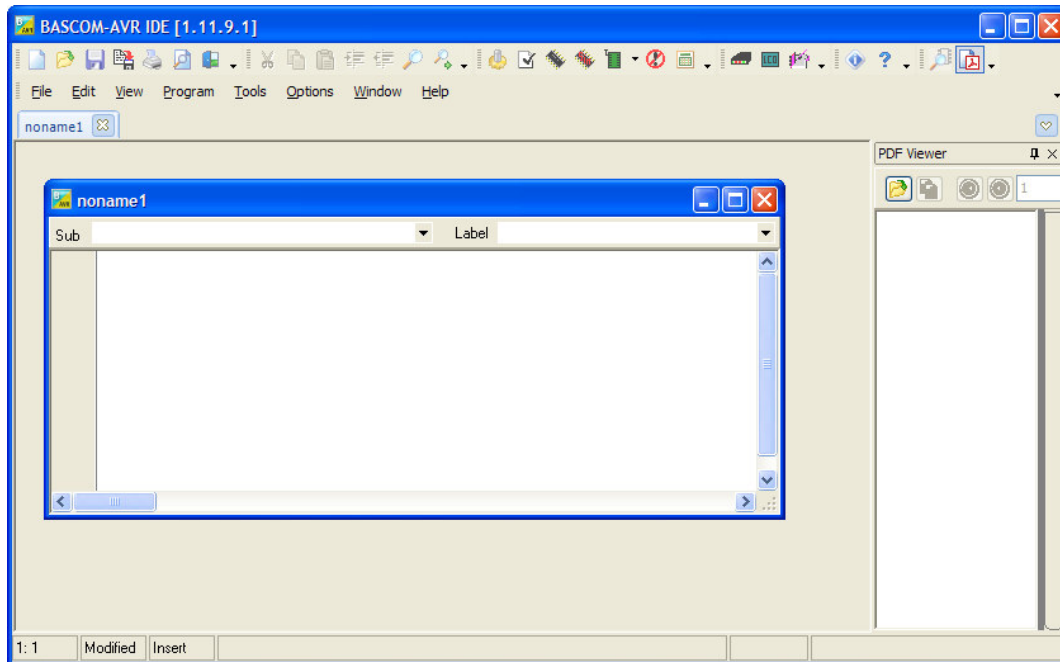
## O Compilador BASCOM

O compilador AVR BASCOM é de fato uma excelente ferramenta para o primeiro contato com o AVR, já que em função de ser uma linguagem de alto-nível diversas funções já estão prontas, como a de comunicação serial, display LCD, display gráfico dentre outras o que de fato irá estimular o leitor a conhecer cada vez mais ambas as ferramentas. Para baixar a versão DEMO deste software, que compila até 4kB de código você pode acessar o site [WWW.mcselec.com](http://WWW.mcselec.com) e baixar o mesmo gratuitamente na seção Download do site.

Após o download e instalação do BASCOM, inicie o mesmo, teremos a seguinte tela:



Para criar um novo programa, vá no menu File e em seguida New. Agora a seguinte tela será apresentada:



É nesta janela branca que foi aberta (área de edição) que iremos escrever nossos projetos em BASIC. A forma básica para escrever um programa em BASIC é a seguinte:

```
$regfile = "m8def.dat"
```

```
$crystal = 1000000
```

```
$baud = 1200
```

Vejamos o significado de cada uma destas linhas:

```
$regfile="m8def.dat"
```

Como o BASCOM trabalha com diversos microcontroladores da linha AVR, devemos informar nesta linha com qual modelo iremos compilar o nosso projeto. Neste caso é o "m8def.dat" que é referente ao ATMEGA8.

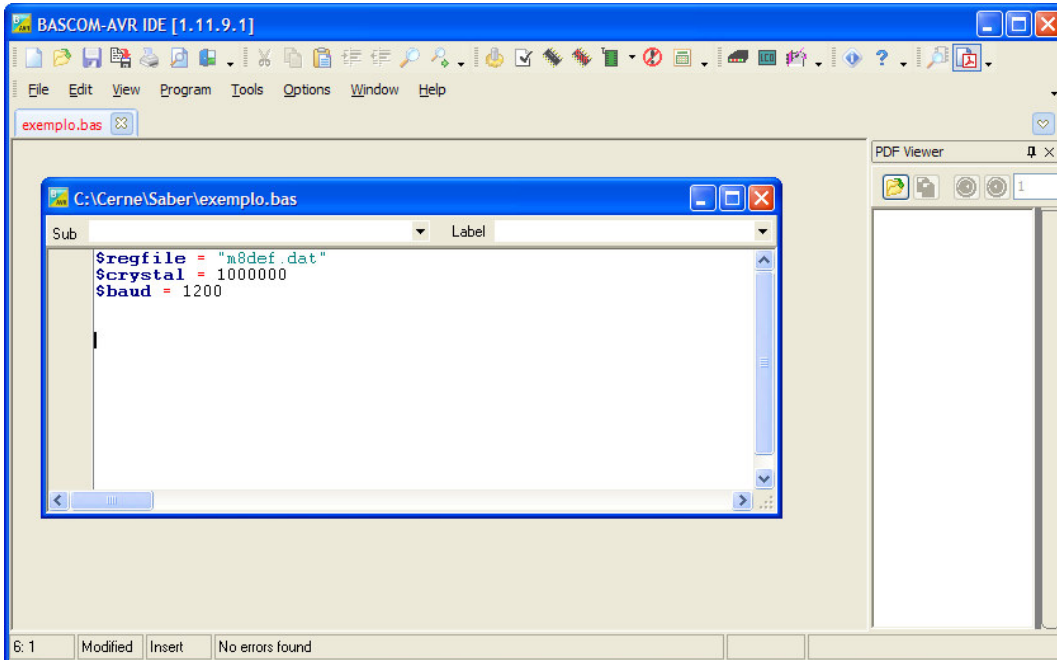
```
$crystal=1200
```

Nesta linha definimos a velocidade de processamento no AVR. O default deste microcontrolador é funcionar com clock interno configurado a 1 MHz. Por isso nesta linha está configurado 1000000 (um milhão) de Hz, que é referente a 1 MHz. Porém observe que nada impede que você funcione com velocidades maiores, observando que o clock máximo deste microcontrolador são de 16 MHz.

```
$baud=1000000
```

Agora estamos configurando a velocidade que será usada para comunicarmos com o PC, neste caso a velocidade é de 1200 bps (bits por segundo).

O código escrito no BASCOM fica como apresentado na figura abaixo:



```
Sub
$regfile = "m8def.dat"
$crystal = 1000000
$baud = 1200
```

Para compilar este arquivo, ou seja transformar o mesmo em código hex que poderá ser gravado em microcontrolador basta pressionar F7 ou ir no menu Program -> Compile. Neste momento inicia a compilação e caso não seja encontrado nenhum erro na compilação, o arquivo hex será gerado e disponibilizado para gravação do microcontrolador.

## Funções de Comunicação Serial

Para comunicarmos serialmente do microcontrolador com o PC e vice-versa, iremos utilizar uma função do compilador BASCOM chamada *Input*. Veja a sintaxe desta função abaixo:

`Input "Texto a ser impresso no PC"`, variável que receberá conteúdo do PC

Observe que em *"Texto a ser impresso no PC"* informamos o texto que será enviado para o PC e apresentado para o usuário do mesmo. Em *"variável que receberá conteúdo do PC"* está informada a variável que receberá o conteúdo no qual será enviado para a placa, neste caso os dois valores a serem somados e enviados novamente para o PC.

Além da função *input* faremos uso de mais uma função, chamada *print*. Esta função permite que seja enviado para o PC o conteúdo de uma variável ou algum texto, no nosso caso será fundamental para enviar a soma dos dois valores previamente informados.

### Código

Abaixo está apresentado o código do nosso exemplo:

```
$regfile = "m8def.dat"
$crystal = 1000000
$baud = 1200
Dim A As Integer
Dim B As Integer
Dim C As Integer
Repete:

Input "Informe um valor: " , A
Input "Informe um segundo valor: " , B

C = A + B

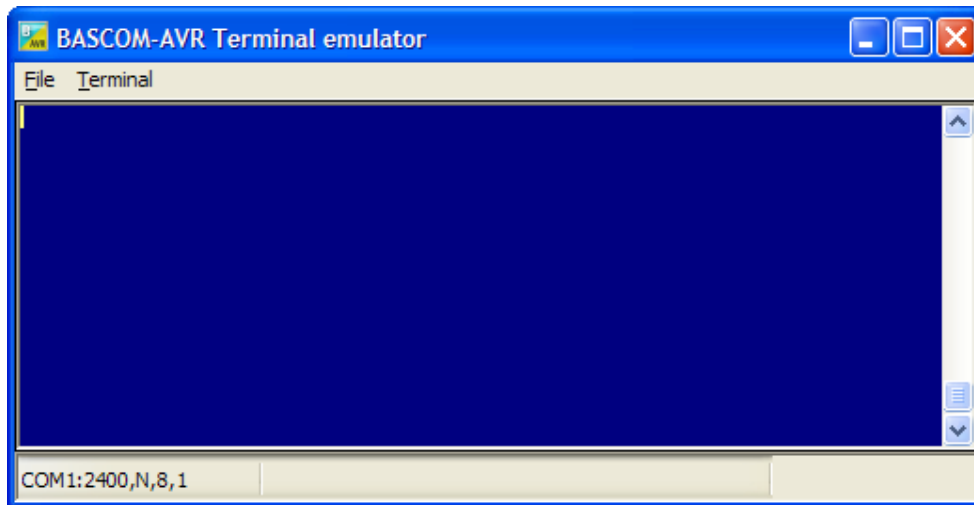
Print C

Goto Repete
```

Vamos entender melhor este código a partir da declaração das variáveis (dim a as integer) já que a parte anterior foi explicada anteriormente. Neste ponto, estamos declarando três variáveis do tipo inteiro (integer) que podem armazenar valores de -32768 a 32767. Elas serão usadas para armazenar o valor enviado do PC para a placa assim como para se achar a soma das duas variáveis. Em seguida, encontramos a função *input* que envia para o PC um texto ("*Informe um valor:* ") solicitando que seja enviado um valor numérico para a placa onde neste caso este valor será armazenado na variável A. Em seguida mesma operação, porém armazenando o valor na variável B. Finalmente, ambos os valores são somados e armazenados na variável C ( $C=A+B$ ). Agora este valor é enviado para o PC através da função *print* (`print c`) em que o valor da soma é enviado para o PC com a soma de A e B. Em seguida o software volta a solicitar as mesmas informações, já que há o comando *goto repete* que faz com que o mesmo volte para a primeira função *input* e repete todo o processo.

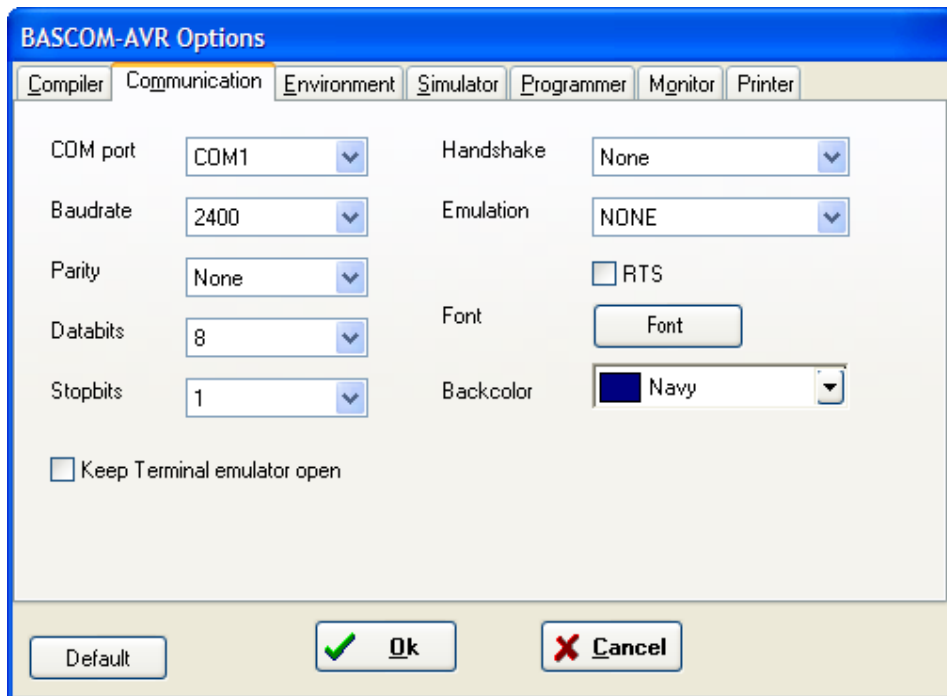
### Testando o Programa

Para testarmos o programa, podemos usar o terminal presente no próprio BASCOM. Para isso vá no menu *Tools -> Terminal Emulator*. Teremos a seguinte tela:



Devemos primeiramente ajustar a velocidade de comunicação do terminal emulador, já que a velocidade de comunicação configurada na placa foi de 1200 bps. Para isso, vá ao menu *Terminal -> Settings*. Teremos acesso a seguinte tela:





No campo Baud Rate, ajuste a velocidade para 1200 e em seguida pressiona Ok. Agora é só testar o programa pelo Terminal Emulator.

### Conclusão

Inúmeras são as aplicações que podem ser desenvolvidas com o AVR e o BASCOM. Recomendo ao leitor que vasculhe nos exemplos que vem na pasta de exemplos do próprio BASCOM as diversas aplicações que este traz consigo. Em breve tratarei de mais aplicações do AVR com o BASCOM assim como outros compiladores, até breve!