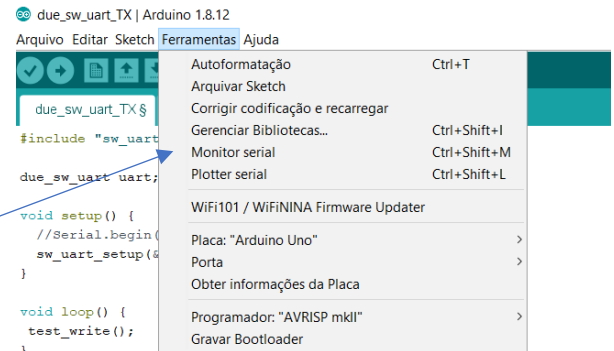


CAMADA FÍSICA DA COMPUTAÇÃO

ENGENHARIA DA COMPUTAÇÃO - Rodrigo Carareto – 0#07E4/01

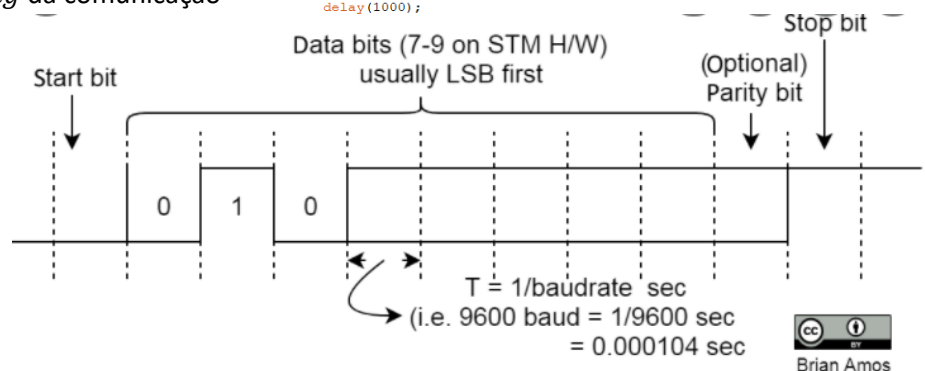
PROJETO 5 – Serialização UART

O objetivo desse projeto é te **desafiar** a produzir a serialização de um byte e enviá-lo através de 1 pino digital qualquer de um Arduino para outro Arduino, que receberá a mensagens no padrão UART através de outro pino digital qualquer. Obviamente você não poderá usar os pinos *tx* e *rx* dos Arduinos. A ideia é que você construa um algoritmo que produza a saída UART em um pino digital genérico. Para isso, você deverá codificar o caractere através da tabela ASCII e enviar os bits de acordo com frame UART com 1 bit de paridade, 1 *start*, 1 *stop* bit e um certo *baudrate*. No arduino que receberá o byte enviado, você poderá observar se o recebimento ocorreu através do *monitor serial* do Arduino receptor.



Lembre-se de como funciona o *framing* da comunicação UART:

Você deve gerar um código que produza num pino do Arduino a serialização correta de um frame UART respeitando os tempos corretos entre os bits. Se fizer tudo corretamente, seu caractere será transmitido via transmissão serial UART e você não estará usando o *chip* UART do Arduino!



Programando em Arduino

Você precisará aprender, caso não saiba, como:

- Definir pinos digitais de escrita e leitura no arduino.
- Habilitar a saída serial para monitoramento de uma variável.
- Como construir um byte representando o caractere, a partir de leitura de cada bit! Para isso, entenda o funcionamento dos operadores "<<" (left shift) e também do |= (compound bitwise or)
- Como produzir cada bit na saída digital a partir de leitura de um byte! Para isso, entenda o funcionamento dos operadores ">>" (right shift) e também do & (compound bitwise and).
- Não se esqueça que a leitura dos níveis lógicos no receptor não pode ser feitas nos instantes de transição do nível lógico (como comentado em aula).
- Escrever esse caractere na saída serial.

Caso você ainda não tenha tido experiência de programação em Arduino, procure tutoriais para os primeiros passos, como por exemplo:

<https://www.circuitar.com.br/tutoriais/programacao-para-arduino-primeiros-passos/>

<https://www.makerhero.com/blog/primeiros-passos-com-arduino/>



CAMADA FÍSICA DA COMPUTAÇÃO

ENGENHARIA DA COMPUTAÇÃO - Rodrigo Carareto – 0#07E4/01

<https://www.arduino.cc/reference/pt/language/functions/digital-io/digitalwrite/>

<https://www.arduino.cc/reference/pt/language/functions/digital-io/digitalread/>

Exemplos de códigos

Função de espera

Você poderá fazer a função espera utilizando loop com a função de asm("NOP")

```
67 // MCK 21MHz
68 void _sw_uart_wait_half_T(due_sw_uart *uart) {
69     for(int i = 0; i < 1093; i++)
70     //for(int i = 0; i < 8333; i++) //clock de 16M
71     asm("NOP");
72 }
```

Cálculo de paridade

Você poderá utilizar funções para calcular a paridade baseadas em bit shift e o operador and:

```
int calc_even_parity(char data) {
    int ones = 0;

    for(int i = 0; i < 8; i++) {
        ones += (data >> i) & 0x01;
    }

    return ones % 2;
}
```

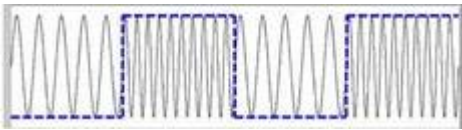
Escrita no monitor serial para conferência

Utilize a saída serial do Arduino conectada ao seu computador para printar a variável recebida.

```
void test_write() {
    sw_uart_write_string(&uart, "B");
    delay(10);
}
```

Montando o byte recebido:

Você deverá usar a estratégia de bit shift e o operador or para compor o byte recebido.



CAMADA FÍSICA DA COMPUTAÇÃO

ENGENHARIA DA COMPUTAÇÃO - Rodrigo Carareto – 0#07E4/01

```
for(int i = 0; i < databits; i++) {  
    aux |= digitalRead(pin_rx) << i;  
    _sw_uart_wait_T(uart);  
}
```

Montagem

Você precisará de 2 Arduinos. Um deles irá funcionar como o transmissor de um caractere qualquer, em loop, utilizando para isso o padrão UART. Você deverá então conectar um dos pinos digitais de um Arduino a outro pino digital do segundo Arduino. Não esqueça de que agora os botões resets não deverão mais estar aterrados, pois você estará usando o processador do Arduino. Lembre-se também de que você terá conectar os terras dos dispositivos.

Analog Discovery

Você deverá utilizar o Wave Forms para analisar a transmissão. Para isso, você irá conectar uma das entradas digitais do Analog Discovery ao pino que está produzindo o sinal UART. No menu do software Wave Forms você irá selecionar o analisador lógico e configurar um canal UART. Você poderá observar então, ao configurar uma escala de tempo adequada, os bits que compõem o frame UART e o caractere transmitido!

Link de ajuda: <https://digilent.com/blog/uart-explained/>

Entrega

Como entrega você poderá apresentar para seu professor em a transmissão.

C -> Se você fizer a comunicação em um baudrate qualquer.

B -> C + Identificação de erro de paridade.

B+ -> B + Se você fizer a transmissão em um baudrate de 9600 bits/s

A -> B+ + Não utilizar função delay ou sleep.

A+ -> monitorar com Analog Discovery.

A data limite para a apresentação será dia 10/10.