

Projeto UART

Lucas Chen Alba

1) Qual o significado da sigla UART? Explique cada um dos termos. Qual a diferença entre uma transmissão síncrona e uma transmissão assíncrona?

R: UART (Universal Asynchronous Receiver/Transmitter) é um tipo de transmissão em que um periférico de um microcontrolador converte bytes em bits a serem enviados/recebidos.

Para a transmissão assíncrona ocorrer, é necessário que o receptor e o transmissor estejam à um mesmo baud rate, assim o receptor saberá o "tempo" de 1 bit. Já na síncrona, tudo é feito a partir de um clock, que precisa ser enviado através de outra linha de comunicação porém não necessita que o receptor saiba o baud rate previamente.

A transmissão síncrona também suporta muito mais protocolos e altas taxas de transmissão (baud rate)

2) Qual a função dos bits de paridade, start e stop?

R: O bit de paridade é um método bastante simples de checagem de erros em dados recebidos, ele funciona a partir da contagem de bits "1" de um payload, caso a paridade seja ímpar e a contagem seja ímpar, adiciona-se um bit "1" ao final do pacote, ou do payload (dependendo do que queremos fazer a checagem), caso a paridade seja par e a contagem seja ímpar, adicionamos um bit "0".

3) Mostre os valores no código (pode ser prints de tela) "setados" para esses bits (paridade, start, stop)". Mostre uma através do Analog Discovery um "frame" de mensagem onde pode-se identificar esses 3 bits.

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

    for (int i = 0; i < 8; i++) {
        ones += (data >> i) & 0x01;
    }
    return ones % 2;
}
```

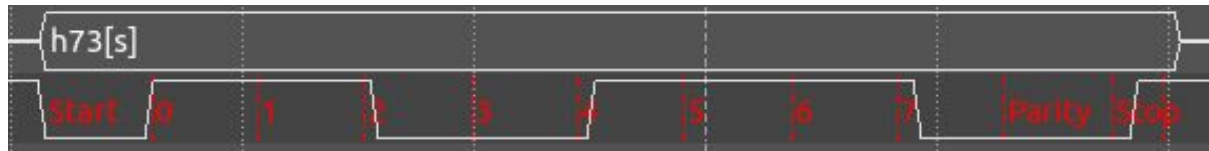
Esta função é a que calcula a paridade dos dados a serem enviados. Ela é implementada no seguinte trecho da função `void sw_uart_write_byte(due_sw_uart *uart, char data)`:

```
// send parity
int parity = 0;
if (uart->paritybit == SW_UART_EVEN_PARITY) {
    parity = !calc_even_parity(data);
}
else if (uart->paritybit == SW_UART_ODD_PARITY) {
    parity = calc_even_parity(data);
}
```

O start bit e stop bit são enviados nos seguintes trechos da mesma função citada:

```
// send start bit
digitalWrite(uart->pin_tx, LOW);
_sw_uart_wait_T(uart);
```

```
// send stop bit
for (int i = 0; i < uart->stopbits; i++) {
    digitalWrite(uart->pin_tx, HIGH);
    _sw_uart_wait_T(uart);
}
```



4) Envie e recepcione mensagens entre dois arduinos através das funções de transmissão UART. Provoque erros de "frame" e paridade. Mostre isso acontecendo.

Podemos provocar erros de paridade alterando a seguinte linha quando estamos implementando a função `calc_even_parity(data)`:

```
parity = !calc_even_parity(data);
```

Para:

```
parity = calc_even_parity(data);
```

E vice-versa caso a paridade seja par. A seguir percebe-se vários pacotes seguidos com este erro.



Para causar erros de framing, alterei a seguinte parte da função `void _sw_uart_wait_half_T(due_sw_uart *uart)`:

```
// MCK 21MHz
void _sw_uart_wait_half_T(due_sw_uart *uart) {
    for (int i = 0; i < 1093; i++)
        asm("NOP");
}
```

Para:

```
// MCK 21MHz
void _sw_uart_wait_half_T(due_sw_uart *uart) {
    for (int i = 0; i < 1200; i++)
        asm("NOP");
}
```

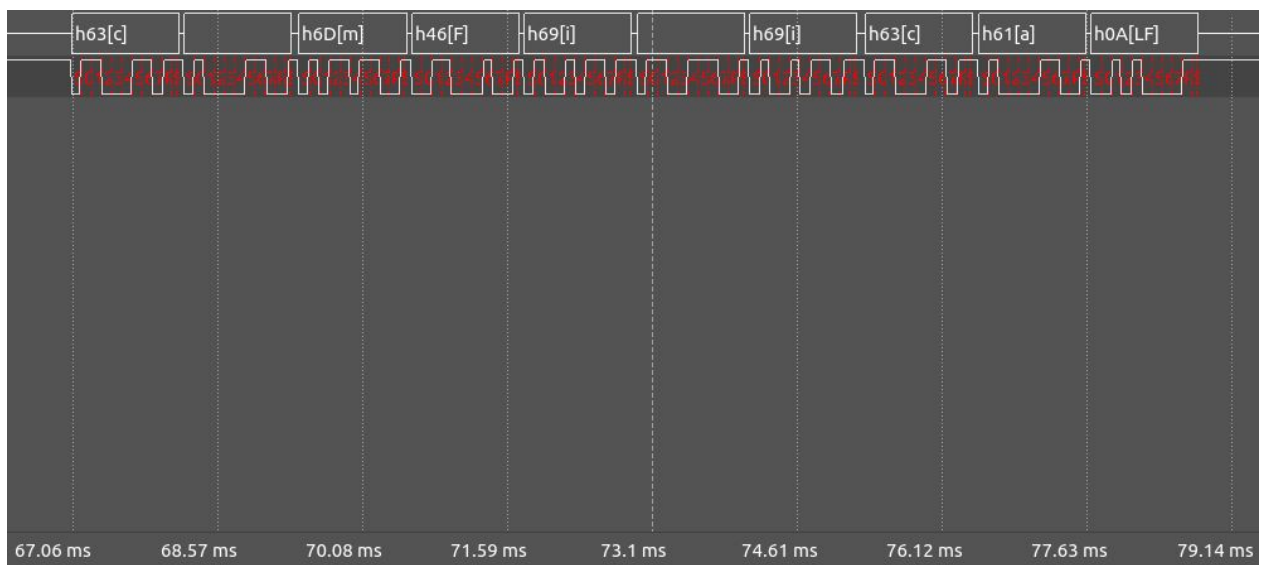
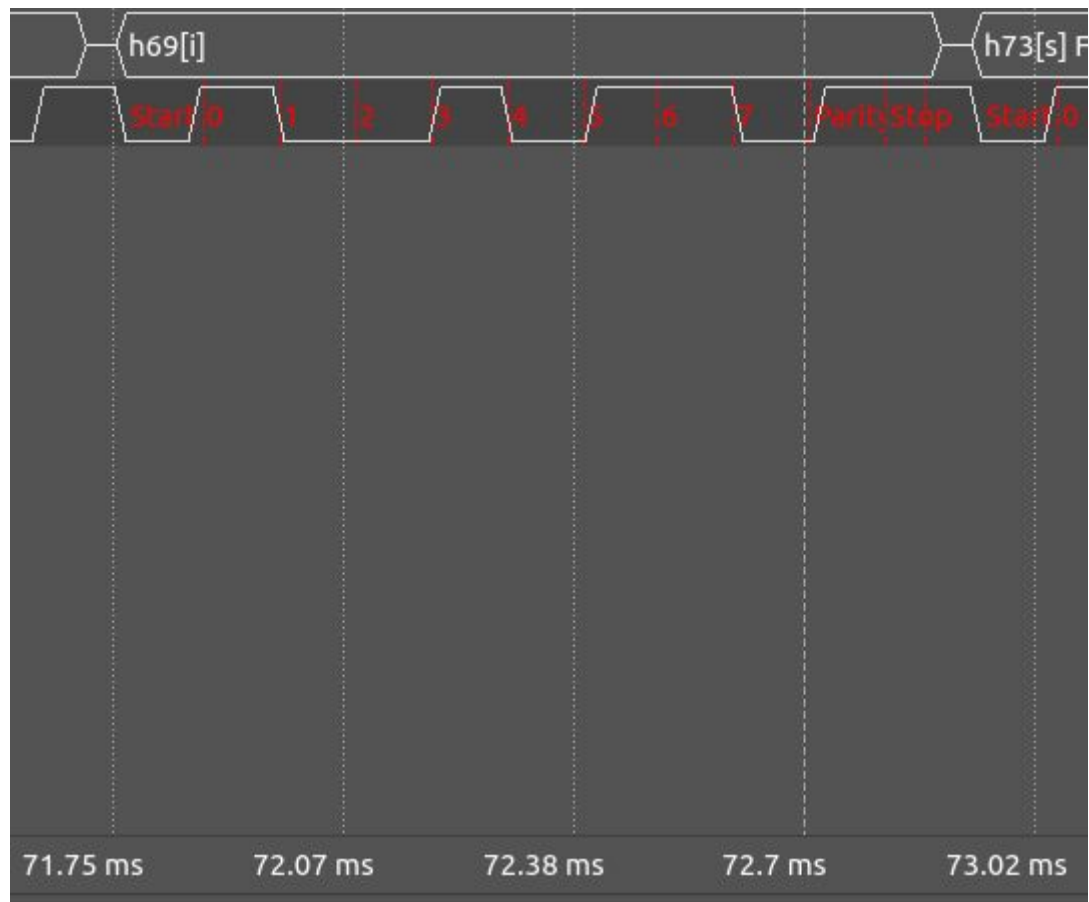
Isso fará com que o intervalo entre cada bit fique substancialmente menor, fazendo com que a UART do receptor capture os sinais em tempos errados, causando não só erros de framing como erros de paridade. A seguir pode-se observar vários pacotes seguidos com erros.



5) Como você alteraria a Baud Rate dessa transmissão? Altere e mostre um print do Analog Discovery "pausado" mostrando o período de um bit para essa velocidade de transmissão.

R: Para mudar o baud rate da transmissão é necessário alterar justamente a parte citada na pergunta anterior, pois alterando a função `void _sw_uart_wait_half_T(due_sw_uart *uart)` estamos alterando a quantidade de “NOP” que o assembly do programa em .cpp escreve entre as mensagens, resultando em uma alteração no tempo entre as escritas (bits).

Aqui tem-se uma imagem de um dos pacotes e da mensagem total, respectivamente. Com Baud Rate original o tempo total de envio de todos os dados é de aproximadamente 0,0111s.



Aqui tem-se uma imagem de um pacote e da mensagem inteira, respectivamente. O Baud Rate foi alterado para metade do original. Percebe-se que há menos variações na linha branca,

significando que houve um recebimento de menos dados. Tendo como período aproximadamente 0,027s.

