

Transmissão Serial Assíncrona

Versão LabEAD 2020

RESUMO

O objetivo desta experiência é projetar circuitos digitais para comunicação serial de dados (transmissão de dados) com um terminal de dados, utilizando a norma EIA-RS-232C e o código ASCII (*American Standard Code Information Interchange*). A parte experimental consiste no projeto e na implementação de um circuito digital em VHDL que envia dados digitais (caracteres em código ASCII) para um terminal serial usando a linguagem de descrição de hardware VHDL e uma placa de desenvolvimento FPGA DE0-CV da Altera.

OBJETIVOS

Após a conclusão desta experiência, os seguintes tópicos devem ser conhecidos pelos alunos:

- Comunicação serial assíncrona (RS232C);
- Conversores de nível de tensão;
- Metodologia de Projeto de Circuitos Digitais com FPGA;
- Projeto de circuitos com VHDL;
- Depuração com ferramentas.

1. PARTE EXPERIMENTAL

A parte experimental envolve a realização de um conjunto de atividades visando o desenvolvimento de um circuito digital para a transmissão de dados para um terminal serial, usando a placa de desenvolvimento FPGA DE0-CV.

Esta experiência foi planejada para ser executada de forma remota, usando os recursos disponíveis na plataforma LabEAD. A bancada remota no Laboratório Digital inclui computador, uma placa FPGA DE0-CV, um Analog Discovery e os componentes do LabEAD (placa ESP8266/Wemos D1 mini, webcam e outros componentes).

1.1. Atividade 1 – Estudo do Projeto Base do Circuito de Transmissão Serial

Esta atividade envolve o estudo e a familiarização do projeto do **circuito base** de transmissão serial assíncrono, onde cada bit é transmitido a cada borda do sinal de clock. O arquivo `tx_serial_base.qar` fornecido contém os arquivos para o Intel Quartus Prime.

Especificação do Projeto: O circuito fornecido transmite um caractere ASCII especificado na entrada `dados_ascii` com o acionamento do sinal `partida`. Ao final da transmissão, o sinal `pronto` é acionado. A saída `saida_serial` deve ser usada para ser ligada ao terminal serial. A entrada `paridade` define a paridade usada na transmissão. A configuração da comunicação serial adotada no projeto é denominada 7E2, ou seja, 7 bits de dados, paridade par (*even*) e 2 *stop bits*. A figura 1 mostra a interface externa do circuito, com os sinais de entrada e saída.



Figura 1 – Interface do Circuito de Transmissão Serial Assíncrona.

- a) Abrir o arquivo QAR com o Intel Quartus Prime e estudar o código VHDL do circuito. Em seguida:
- Identificar os componentes do fluxo de dados e a composição deles para formar o módulo fluxo de dados (entidade `tx_serial_fd`).
DICA: use a saída da ferramenta *RTL Viewer* para ilustrar este estudo.
 - Analisar a máquina de estados da unidade de controle (entidade `tx_serial_uc`) e estude os estados e as transições entre eles.
DICA: use a saída da ferramenta *State Machine Viewer* e edite a figura para incluir as condições de transição de estados e as saídas dos sinais de controle em cada estado.
- Os diagramas gerados são importantes para as modificações a serem realizadas no projeto da Atividade 2 a seguir.
- b) Defina os casos de testes que devem ser executados para assegurar o correto funcionamento do circuito completo. Se julgar necessário, defina alguns sinais de depuração.
- c) Simular o funcionamento do projeto, usando os casos de teste definidos, e anexar as formas de onda obtidas no Planejamento.
- d) Redigir um parágrafo descrevendo textualmente o funcionamento do circuito desde o acionamento do `reset`, espera do acionamento do sinal de `partida`, até o final da transmissão e a ativação do sinal `pronto`.
- e) Os resultados desta atividade devem ser incluídos no Planejamento.

1.2. Atividade 2 – Projeto do Circuito de Transmissão Serial com *clock* de 50MHz

Esta atividade envolve a modificação do projeto base para transmissão serial assíncrona desenvolvido na Atividade 1 para usar o *clock* de 50MHz da placa DE0-CV. Esta modificação deve criar um novo projeto no Intel Quartus Prime denominado `tx_serial_tick`.

O uso de um sinal de *clock* cuja frequência é superior à taxa de transmissão serial pode ser implementado pelo uso da **técnica de superamostragem** do sinal serial, ou seja, cada *bit* do dado serial é transmitido após uma sequência de bordas do *clock*. A figura 2 ilustra a aplicação desta técnica para a transmissão serial.

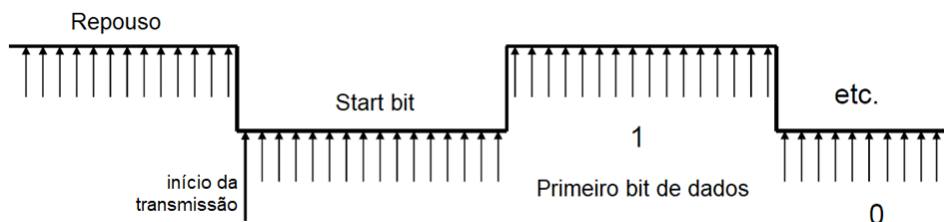


Figura 2 – Transmissão serial com superamostragem e ocorrência das bordas de *clock*.

O momento em que cada *bit* deve ser transmitido pode ser definido pelo circuito digital usando um contador auxiliar denominado **gerador de tick**. Um *tick* ocorre a cada intervalo de *bit*, informando o circuito de transmissão para mudar para o próximo dado serial. Por exemplo, para uma frequência de *clock* de 50MHz (período de 20ns) e uma taxa de transmissão de 115.200 *bauds*, cada bit deverá ser transmitido a cada por 8,68µs ou 434 ciclos de *clock* ($50.000.000/115.200$). Assim, o gerador de tick pode ser implementado usando a saída de indicação de fim de contagem de um contador síncrono. A figura 3 apresenta as formas de onda necessárias para a implementação da superamostragem para a transmissão serial.

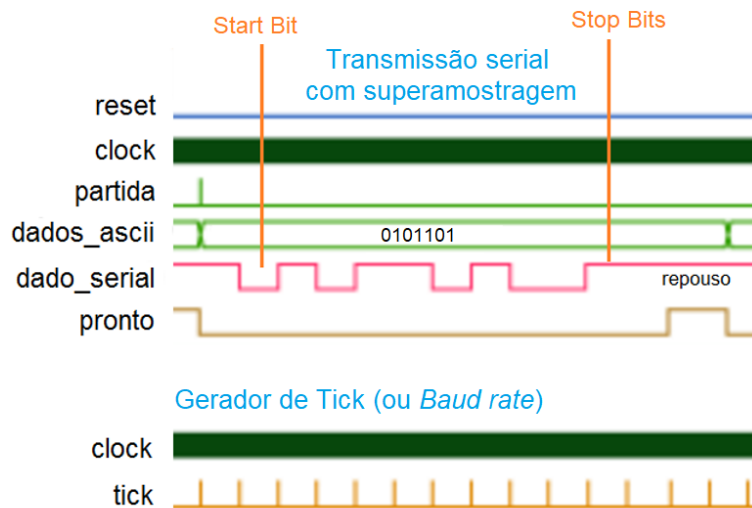


Figura 3 – Principais sinais para a transmissão serial com superamostragem.

A implementação da superamostragem pode ser realizada com a modificação da máquina de estados da unidade de controle, de forma a incluir um estado de espera de ocorrência do sinal de *tick*. A figura 4 apresenta o diagrama de transição de estados. Compare com o diagrama de transição de estados do projeto base, estudado na Atividade 1.

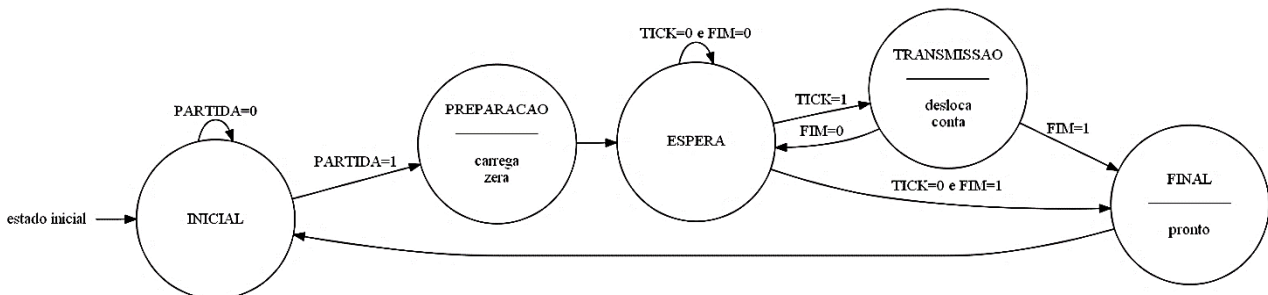


Figura 4 – Diagrama de transição de estados para aplicação da superamostragem na transmissão serial.

- f) Defina as constantes nos diversos componentes VHDL do projeto do circuito para uma taxa de transmissão de 115.200 *bauds* e configuração de transmissão 7E2.
- g) Documente o projeto do grupo. **DICA:** anexe as saídas das ferramentas *RTL Viewer* (figura 5) e *State Machine Viewer* editadas para completar as informações de funcionamento.

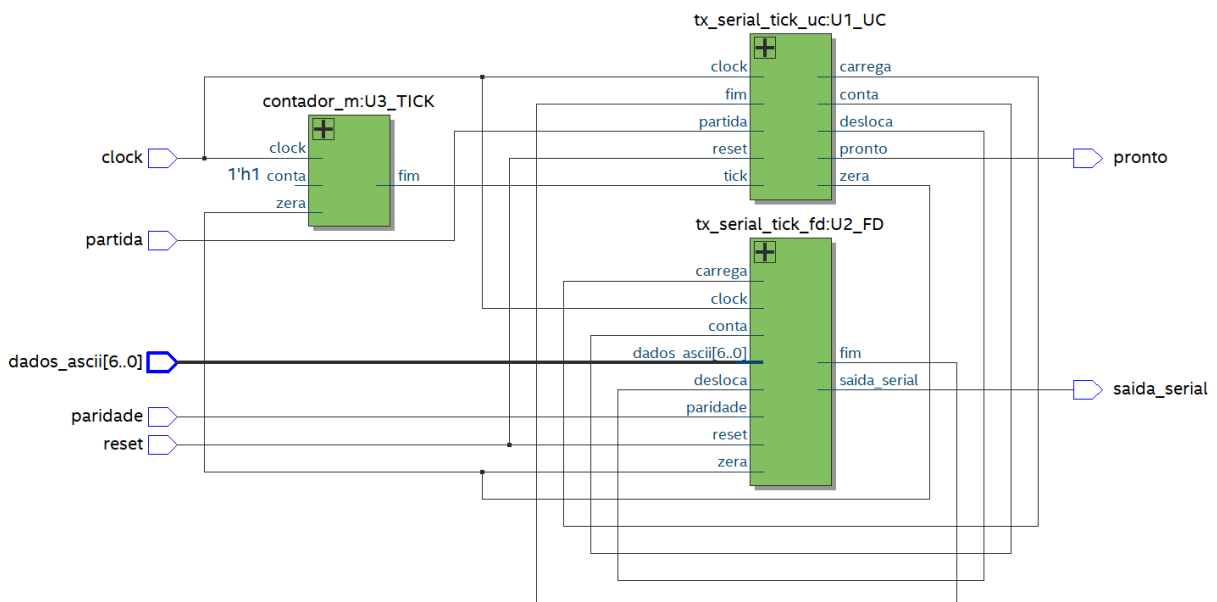


Figura 5 – Exemplo de saída da ferramenta *RTL Viewer* para o fluxo de dados do projeto.

- h) Defina os casos de testes que devem ser executados para assegurar o correto funcionamento do circuito. Se julgar necessário, defina alguns sinais de depuração. Estes sinais de depuração podem ser designados aos *leds* da placa FPGA e monitorados remotamente pela imagem fornecida pela *webcam*.
- i) Simular o funcionamento do projeto e anexar as formas de onda obtidas no Planejamento.
DICA: para simulação do circuito, usar uma relação de 10 *clocks* por *tick* devido à limitação do tempo de simulação no Intel Quartus Prime. Na síntese do circuito final, volte o gerador de *tick* para o fator de divisão real.
- j) O circuito de transmissão serial modificado deve usar a seguinte designação de pinos.

signal	pino	pino FPGA	Blynk	Analog Discovery	
CLOCK	CLK_50	M9	-	-	-
RESET	GPIO_0_D0	N16	V10	-	-
PARTIDA	GPIO_0_D1	B16	V11	-	-
PARIDADE	GPIO_0_D10	N21	V20	-	-
DADOS_ASCII[0..6]	GPIO_0_D9, D8,D7,D6, D5,D4,D3	M21,M20,K22, K21,K20, D17,C16	V80 (<i>slider</i>)	-	-
PRONTO	<i>led</i> LEDR0	AA2	-	-	-
SAIDA_SERIAL * (CH1)	GPIO_1_D27	F15	-	Scope	CH1
SAIDA_SERIAL	GPIO_1_D4	A13	V77 (terminal UART)	-	-

(*) designação a ser usada para verificação da forma de onda.

- k) Submeter o novo arquivo QAR (`exp2_tick_txby.qar`) junto com o Planejamento do grupo.

1.3. Atividade 3 – Implementação do Projeto na Placa FPGA DE0-CV

Neste item vamos implementar, no Laboratório Digital, o projeto do circuito de transmissão serial modificado na placa FPGA DE0-CV.

- l) Inicialmente, designar o sinal de saída do circuito SAIDA_SERIAL para ser ligado na ferramenta Scope (osciloscópio) do Analog Discovery.
- m) Programar o circuito modificado na placa DE0-CV, usando a infraestrutura de acesso remoto do LabEAD.
- n) Execute o teste de funcionamento do circuito para a taxa de 115.200 *bauds*, aplicando os casos de teste definidos no Planejamento.
- o) Use a ferramenta *Scope* do **Analog Discovery** para a visualização do sinal de saída serial para fins de depuração. Adicione figuras com as saídas das ferramentas no Relatório.
DICA: Para esta atividade, é preciso usar os recursos de acesso à bancada remota via AnyDesk.
- p) Se as formas de onda forem validadas, mude a designação do sinal de saída para o pino da GPIO ligado no ESP8266.
- q) Programe o circuito modificado na placa DE0-CV.
- r) Execute o teste de funcionamento do circuito, aplicando mais um a vez os casos de teste definidos no Planejamento.
- s) Verifique o funcionamento do circuito, observando as saídas apresentadas no Terminal remoto do projeto no Blynk.
- t) Colete imagens da tela do projeto do Blynk e documente o funcionamento do projeto no Relatório.
- u) Submeter o arquivo QAR do projeto (`exp2_tick_txby.qar`) junto com o Relatório.

1.4. Atividade 4 –Desafio

Neste item deverão ser estudados e implementados alguns melhoramentos no circuito de transmissão serial na placa DE0-CV.

- v) Uma **modificação** no projeto do circuito de transmissão serial assíncrona será proposta pelo professor. Estude esta modificação e verifique qual parte do projeto deve ser alterado.
- w) Implemente o circuito no Intel Quartus Prime e sintetize-o na placa FPGA DE0-CV. Documente a designação de pinos adotada e alterações no projeto do Blynk.
- x) Documente os resultados obtidos nos testes experimentais realizados.
- y) Submeter o arquivo QAR do projeto do desafio (`exp2_desafio_txby.qar`) junto com o Relatório.

2. BIBLIOGRAFIA

- ALMEIDA, F.V. de; SATO, L.M.; MIDORIKAWA, E.T. **Tutorial para criação de circuitos digitais em VHDL no Quartus Prime 16.1**. Apostila de Laboratório Digital. Departamento de Engenharia de Computação e Sistemas Digitais, Escola Politécnica da USP. Edição de 2017.
- ALTERA. **DE0-CV User Manual**. 2015.
- ALTERA. **Quartus Prime Introduction Using VHDL Designs**. 2016.
- ALTERA. **Quartus Prime Introduction to Simulation of VHDL Designs**. 2016.
- CCITT - Fifth Plenary Assembly. Green Book. Vol. VIII, Geneve, December 1972.
- D'AMORE, R. **VHDL - descrição e síntese de circuitos digitais**. 2ª edição, LTC, 2012.
- Electronic Industries Association. **Interface Between Data Terminal Equipment and Data Communication Equipment Employing Serial Date Interchange EIA-RS-232-C**, Washington, August 1969.
- HELD, G. **Understanding Data Communications**. 6th ed., New Riders, 1999.
- MIDORIKAWA, E.T. **Metodologia de Projeto com Dispositivos Programáveis**. Apostila de Laboratório Digital. PCS-EPUSP, 2016.
- PCS-EPUSP. **Conceitos de Comunicação Serial Assíncrona**. Apostila de Laboratório Digital. 2019.
- TOCCI, R. J.; WIDMER, N.S.; MOSS, G.L. **Sistemas Digitais: Princípios e Aplicações**. Prentice-Hall, 11ª ed., 2011.
- WAKERLY, John F. **Digital Design Principles & Practices**. 4th edition, Prentice Hall, 2006.

3. EQUIPAMENTOS NECESSÁRIOS

- 1 computador com software Intel Quartus Prime.
- 1 dispositivo Analog Discovery da Digilent.
- 1 placa de desenvolvimento FPGA DE0-CV com o dispositivo Cyclone V 5CEBA4F23C7N.
- 1 kit Lab do LabEAD (para implementação do acesso remoto).

Histórico de Revisões

E.S.G. e F.N.A/2001 – revisão
 E.T.M./2004 – revisão
 E.T.M./2005 – revisão
 E.T.M./2008 – revisão
 E.T.M./2011 – revisão
 E.T.M./2012 – revisão
 E.T.M./2013 – revisão da parte experimental
 E.T.M./2014 – revisão
 E.T.M./2015 – revisão do texto
 E.T.M./2016 – revisão
 E.T.M./2017 – revisão
 E.T.M./2018 – revisão
 E.T.M./2019 – revisão
 E.T.M./2020 – revisão e reorganização da experiência para acesso remoto.

Apêndice:

Configuração do Projeto do Blynk

Use o projeto do Blynk criado na experiência anterior, de forma a reutilizar o Auth Token. Inicialmente, remova o *widget* de botão, selecionando o objeto e deslizando até o ícone de reciclagem. A figura 6 ilustra esta etapa.

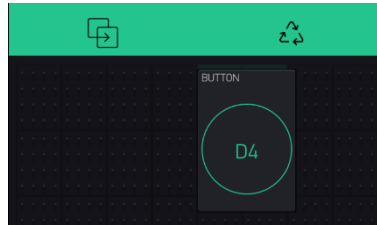


Figura 6 – Reciclagem do botão da experiência anterior.

O projeto do Blynk para esta experiência inclui um conjunto de *widgets* apresentado na figura 7 abaixo.

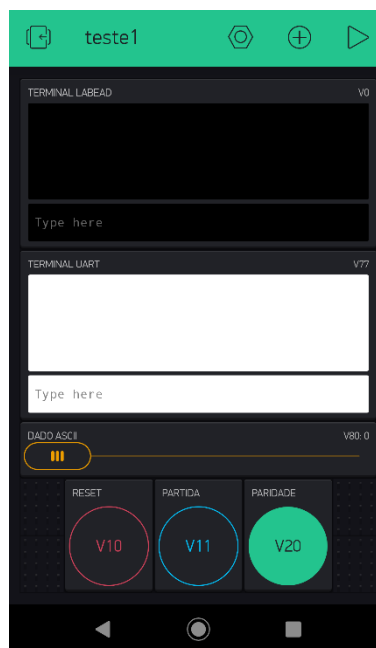


Figura 7 – Imagem da tela do projeto do Blynk para a experiência de Transmissão Serial Assíncrona.

A tabela 1 abaixo apresenta uma relação dos *widgets* e respectivas configurações.

Tipo de <i>widget</i>	Identificação	Pino associado	Configuração do widget
Terminal	Terminal LabEAD	V0	Cor escura
Terminal	Terminal UART	V77	Cor clara
Slider	Dado ASCII	V80	Intervalo de valores: 0 a 127 Decimals: # Cor: laranja
Botão	Reset	V10	0→1 / Switch Cor: vermelho
Botão	Partida	V11	0→1 / Switch Cor: azul
Botão	Paridade	V20	0→1 / Switch Cor: verde

A figura 7 abaixo resume as configurações dos *widgets* do projeto.

Prestem atenção nas configurações de cada elemento, pois é muito importante que não haja enganos para garantir o pleno funcionamento do projeto.

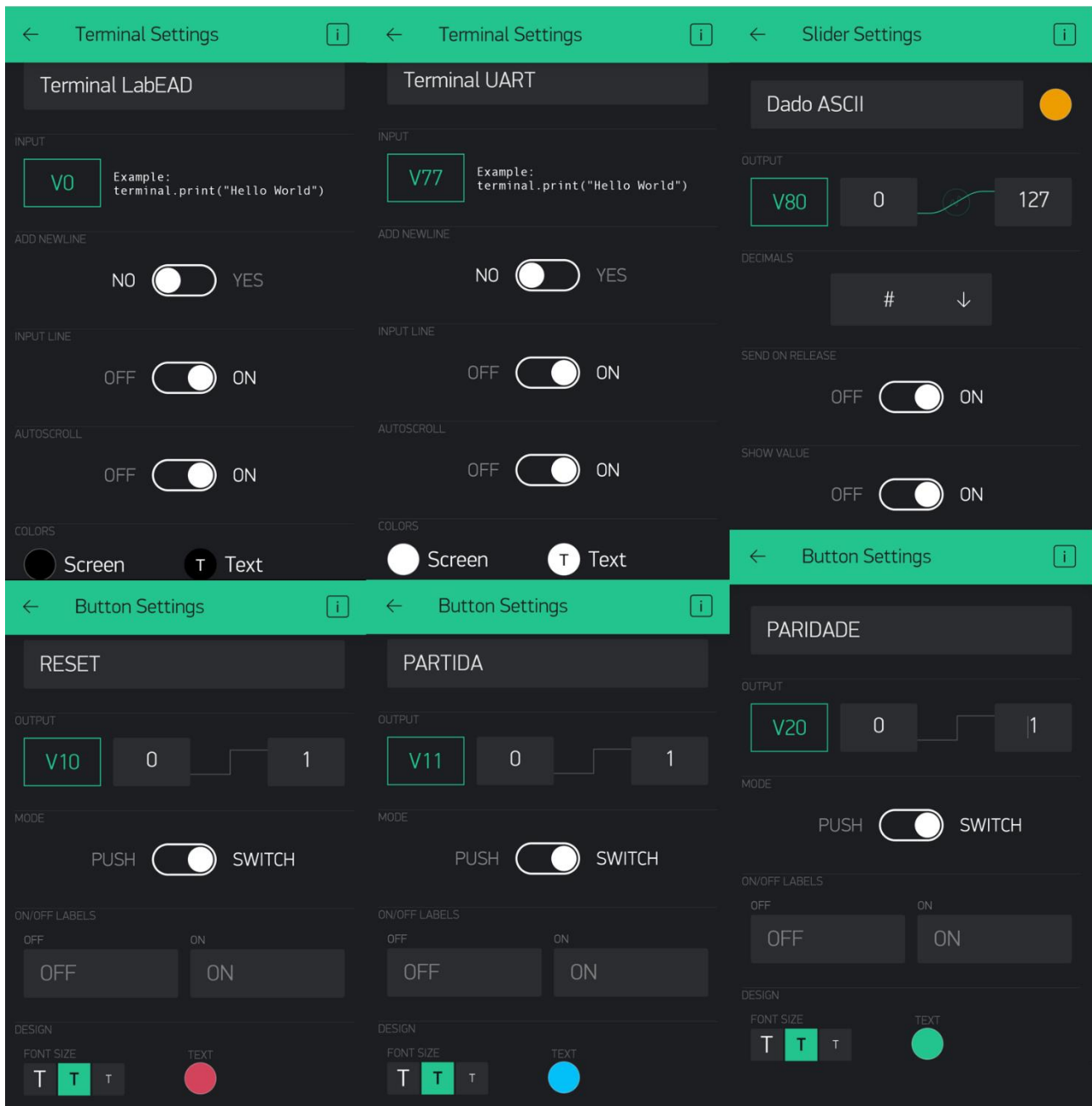


Figura 7 – Resumo das configurações dos *widgets* usados na experiência de Transmissão Serial Assíncrona.