

第1回輪講資料

4321 野秋 琳太郎

2025 年 5 月 23 日

指導教員 宮田 尚起

1. はじめに

LED で光通信をして,「糸なし糸電話」をやりたい. 産技祭にだして, 稲毛研究室と距離で競いたい.

2. 方針

図1のような構成で通信を行うことにした. LED の照度でアナログなデータを送るのは難しいと考えたので,LED の点灯/消灯でデジタルなデータを送ることにした.

デジタル通信には UART を使用する. 信号線が1本でよく,TeraTermなどを利用すればデバッグも容易なためである. midi など, 音を送る通信での採用例もある.

ここで, 光通信はホットカップラとして見做すことができる. よって, 受信側は入力のアナログ信号を UART で送信, 送信側は受信した UART を D-A コンバータからアナログ信号として出力というように, 適当なライブラリを利用すれば簡単に実装できる.

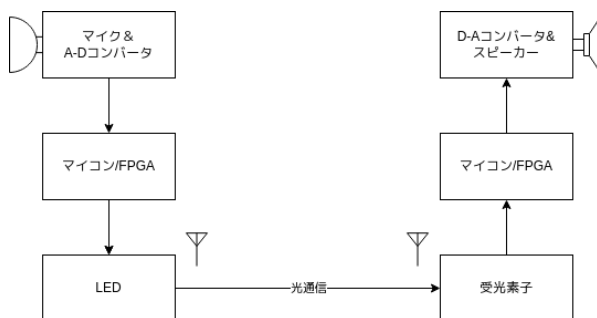


図 1. 光通信のブロック図

3. 分担

班で役割分担をした. その内容は以下のとおりである. 輪講では, 自分の分だけ進

捗を発表する.

- 瀬田: オーディオ入力,A-D 変換
- 本橋:LED を駆動
- 林 :LED, アンテナ
- 野秋:UART 受信,D-A 変換, オーディオ出力

4. 製作

UART の受信,D-A 変換, オーディオ出力には図2にある FPGA,Tang Primer 20k を使用する. 図3に示すとおり,Tang Primer 20k の Dock には DAC とアンプがついており, 外付けの回路を作らずに D-A 変換, オーディオ出力ができる. よって,FPGA で作るべき回路は UART 受信,DAC 用のインターフェースである. 今週は UART の送受信回路を System Verilog で書いた. 1 文字の入出力ができることを確認してある.

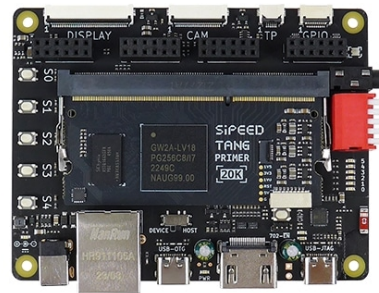


図 2. Tang Primer 20k


```

22 } uart_state;
23
24 //Tx_baud gen
25 reg [11:0]Tx_counter;
26 wire Tx_trig;
27 //uart Tx
28 uart_state Tx_state;
29 logic [2:0]Tx_bit_counter;
30
31 //Rx_baud gen
32 reg [11:0]Rx_counter;
33 wire Rx_trig;
34 wire Rx_valid;
35 //Rx_strt
36 logic Rx_prev;
37 wire Rx_strt;
38 //uart Rx
39 uart_state Rx_state;
40 logic [2:0]Rx_bit_counter;
41
42 //Tx_baud gen
43 always @(posedge clock or negedge reset) begin
44     if(!reset)Tx_counter<=0;
45     else begin
46         Tx_counter<=(baud_div==Tx_counter)?(1'b0):(Tx_counter+1'b1);
47         if(!enable_trig) Tx_counter<=0;
48     end
49 end
50
51 assign Tx_trig=(baud_div==Tx_counter)?(1):(0);
52
53 //uart Tx
54 always @(posedge clock or negedge reset) begin
55     if(!reset)begin
56         Tx_state<=idle;
57         Tx_bit_counter<=0;
58         TxD<=1;
59     end
60     else begin
61         case(Tx_state)
62             idle:begin
63                 TxD<=1;
64                 if(!enable_trig)Tx_state<=strt;

```

```

65         end
66         strt:begin
67             TxD<=0;
68             if(Tx_trig)begin
69                 Tx_state<=trns;
70                 Tx_bit_counter<=0;
71             end
72         end
73         trns:begin
74             TxD<=Tx_buffer[Tx_bit_counter];
75             if(Tx_trig)begin
76                 Tx_bit_counter<=Tx_bit_counter+1'b1;
77                 if(Tx_bit_counter==3'b111) Tx_state<=stop;
78             end
79         end
80         stop:begin
81             TxD<=1;
82             if(Tx_trig)Tx_state<=idle;
83         end
84     endcase
85 end
86 end
87
88 //Rx_baud gen
89 //Rx_trig,Rx_valid
90 always @(posedge clock or negedge reset) begin
91     if(!reset)Rx_counter<=0;
92     else Rx_counter<=(Rx_strt==1 ||
93         baud_div==Rx_counter)?(1'b0):(Rx_counter+1'b1);
94 end
95 assign Rx_trig=(baud_div==Rx_counter)?(1):(0);
96 assign Rx_valid=(valid==Rx_counter)?(1):(0);
97
98 //Rx_strt
99 always@(posedge clock or negedge reset)begin
100     if(!reset) Rx_prev<=1;
101     else Rx_prev<=RxD;
102 end
103
104 //待機状態で,Rxの立下りエッジにいるときRx_strt=1
105 assign Rx_strt=(Rx_state==idle && Rx_prev==1 && RxD==0)?(1):(0);
106

```

```

107 //uart Rx
108 always @(posedge clock or negedge reset) begin
109     if(!reset)begin
110         Rx_state<=idle;
111         Rx_buffer<=0;
112     end
113     else begin
114         case(Rx_state)
115             idle:if(Rx_strt)Rx_state<=strt;
116             strt:begin
117                 if(Rx_trig)begin
118                     Rx_state<=trns;
119                     Rx_bit_counter<=0;
120                 end
121             end
122             trns:begin
123                 //7はマジックナンバー,よくない
124                 if(Rx_valid)begin
125                     Rx_buffer[Rx_bit_counter]<=RxD;
126                     Rx_bit_counter<=Rx_bit_counter+1'b1;
127                     if(Rx_bit_counter==7) Rx_state<=mrgn;
128                 end
129             end
130             mrgn:if(Rx_trig)Rx_state<=stop;
131             stop:if(Rx_trig)Rx_state<=idle;
132         endcase
133     end
134 end
135 endmodule

```

5. 今後の定

D-A コンバータのインターフェースを 作って,手元で確認ができるようにする.