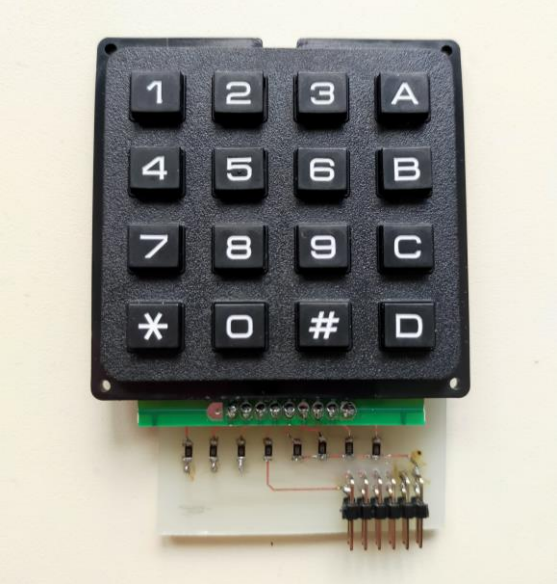
KOCAELİ UNIVERSITY ENGINEERING FACULTY

PYMOD KEYPAD
REFERENCE DOCUMENT



ZEYNEP ÖZÇELİK

MURAT SELİM HAN GEDİK

ZEKİYE HAYTA

2021-2022

DEPARTMENT: Electronic and Communacition Engineering

TEACHER: Phd. ANIL ÇELEBİ

TABLE OF CONTENTS

Functional Description :

The PmodKYPD utilizes 4 rows and columns to create an array of 16 momentary pushbuttons. By driving the column lines to a logic level low voltage one at a time, users may read the corresponding logic level voltage on each of the rows to determine which button, if any, is currently being pressed. Simultaneous button presses can also be recorded, although it is still required to step through each row and column separately in order to ensure that the pressed buttons do not interfere with each measurement.

Features include:

- 16 momentary push-buttons
- Can detect simultaneous button presses
- Isolated rows and columns
- Small PCB size for flexible designs 3.4“ × 2.7” (8.6 cm × 6.9 cm)
- 12-pin Pmod port with GPIO interface
- Follows Digilent Pmod Interface Specification Type 1
- Library and example code available in resource center F

Interfacing with the Pmod :

The PmodKYPD communicates with the host board via the GPIO protocol. Each button is placed within a simple voltage divider circuit. When a button is not pressed, a large pull-up resistor maintains a logic level high voltage on each of the row pins. When a column pin is driven to a logic level low voltage and a corresponding button is pressed, completing the voltage divider circuit, the row pin will then read a logic level low voltage instead.

Header J1					
Pin	Signal	Description	Pin	Signal	Description
1	COL4	Column 4	7	ROW4	Row 4
2	COL3	Column 3	8	ROW3	Row 3
3	COL2	Column 2	9	ROW2	Row 2
4	COL1	Column 1	10	ROW1	Row 1
5	GND	Power Supply Ground	11	GND	Power Supply Ground
6	VCC	Power Supply (3.3V/5V)	12	VCC	Power Supply (3.3V/5V)

Any external power applied to the PmodKYPD must be within a voltage that your system board can handle; It is recommended that Pmod is operated at 3.3V.

Physical Dimensions :

The pins on the pin header are spaced 100 mil apart. The PCB is 3.4 inches long on the sides parallel to the pins on the pin header and 2.7 inches long on the sides perpendicular to the pin header.

MATERIAL LIST

Comment	Description
Component_2	
Header 6X2A	Header, 6-Pin, Dual row
10k	RES SMD 10K OHM 0.05% 1/4W 1206
470	RES SMD 470 OHM 1% 1/4W 1206

VERILOG HDL

TOP.V :

```
// Engineer:
//
// Create Date: 22.12.2021 14:10:49
// Design Name:
// Module Name: Top
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////
module Top(
     clk,
    JA,
    an,
    seg
    );
// ================================================================================================
//                                Port Declarations
// ================================================================================================
    input clk;                  // 100Mhz onboard clock
    inout [7:0] JA;          // Port JA on Nexys3, JA[3:0] is Columns, JA[10:7] is rows
    output [3:0] an;            // Anodes on seven segment display
    output [6:0] seg;           // Cathodes on seven segment display
// ================================================================================================
//                                   Parameters, Regsiters, and Wires
// ================================================================================================
    // Output wires
    wire [3:0] an;
    wire [6:0] seg;

    wire [3:0] Decode;
```

```
// ============================================================================================
//                                          Implementation
// ============================================================================================

    //-----------------------------------------------
    //                          Decoder
    //-----------------------------------------------
    Decoder C0(
            .clk(clk),
            .Row(JA[7:4]),
            .Col(JA[3:0]),
            .DecodeOut(Decode)
    );
    //-----------------------------------------------
    //          Seven Segment Display Controller
    //-----------------------------------------------
    DisplayController C1(
            .DispVal(Decode),
            .anode(an),
            .segOut(seg)
    );
endmodule
```

DECODER.V :

```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date: 22.12.2021 14:10:49
// Design Name:
// Module Name: Decoder
// Project Name:
// Target Devices:
// Tool Versions:
// Desoription:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////
module Decoder(
    clk,
    Row,
    Col,
    DecodeOut
    );
// ============================================================================================
//                                      Port Declarations
// ============================================================================================
    input clk;                      // 100MHz onboard clock
    input [3:0] Row;                // Rows on KYPD
    output [3:0] Col;            // Columns on KYPD
    output [3:0] DecodeOut; // Output data
// ============================================================================================
//                                  Parameters, Regsiters, and Wires
// ============================================================================================
    // Output wires and registers
    reg [3:0] Col;
    reg [3:0] DecodeOut;
```

PG.5

```
    // Count register
    reg [19:0] sclk;
// ================================================================================================
//                                          Implementation
// ================================================================================================
    always @(posedge clk) begin
            // 1ms
            if (sclk == 20'b00011000011010100000) begin
                //C1
                Col <= 4'b0111;
                sclk <= sclk + 1'b1;
            end
            // check row pins
            else if(sclk == 20'b00011000011010101000) begin
                //R1
                if (Row == 4'b0111) begin
                    DecodeOut <= 4'b0001;       //1
                end
                //R2
                else if(Row == 4'b1011) begin
                    DecodeOut <= 4'b0100;       //4
                end
                //R3
                else if(Row == 4'b1101) begin
                    DecodeOut <= 4'b0111;       //7
                end
                //R4
                else if(Row == 4'b1110) begin
                    DecodeOut <= 4'b0000;       //0
                end
                sclk <= sclk + 1'b1;
            end
            // 2ms
            else if(sclk == 20'b00110000110101000000) begin
                //C2
                Col<= 4'b1011;
                sclk <= sclk + 1'b1;
            end
            // check row pins
            else if(sclk == 20'b00110000110101001000) begin
                //R1
```

PG.6

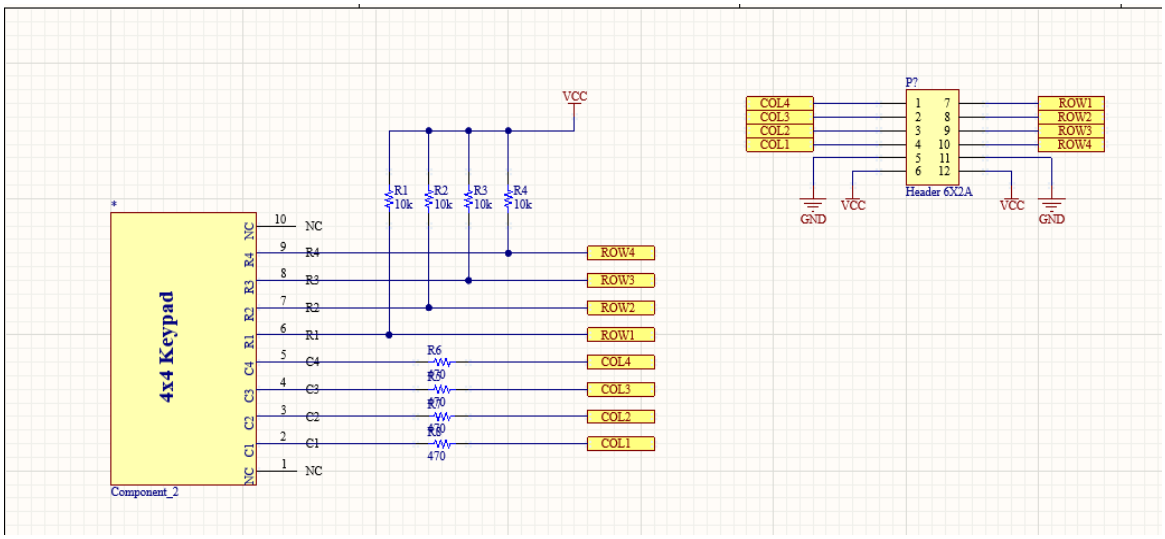
```
        if(Row == 4'b0111) begin
            DecodeOut <= 4'b0011;       //3
        end
        //R2
        else if(Row == 4'b1011) begin
            DecodeOut <= 4'b0110;       //6
        end
        //R3
        else if(Row == 4'b1101) begin
            DecodeOut <= 4'b1001;       //9
        end
        //R4
        else if(Row == 4'b1110) begin
            DecodeOut <= 4'b1110;       //E
        end
        sclk <= sclk + 1'b1;
    end
    //4ms
    else if(sclk == 20'b01100001101010000000) begin
        //C4
        Col<= 4'b1110;
        sclk <= sclk + 1'b1;
    end

    // Check row pins
    else if(sclk == 20'b01100001101010001000) begin
        //R1
        if(Row == 4'b0111) begin
            DecodeOut <= 4'b1010; //A
        end
        //R2
        else if(Row == 4'b1011) begin
            DecodeOut <= 4'b1011; //B
        end
        //R3
        else if(Row == 4'b1101) begin
            DecodeOut <= 4'b1100; //C
        end
        //R4
        else if(Row == 4'b1110) begin
            DecodeOut <= 4'b1101; //D
```

```
                end
                sclk <= 20'b00000000000000000000;
            end
            // Otherwise increment
            else begin
                sclk <= sclk + 1'b1;
            end
    end
endmodule
```

DISPLAY CONTROLLER.V :

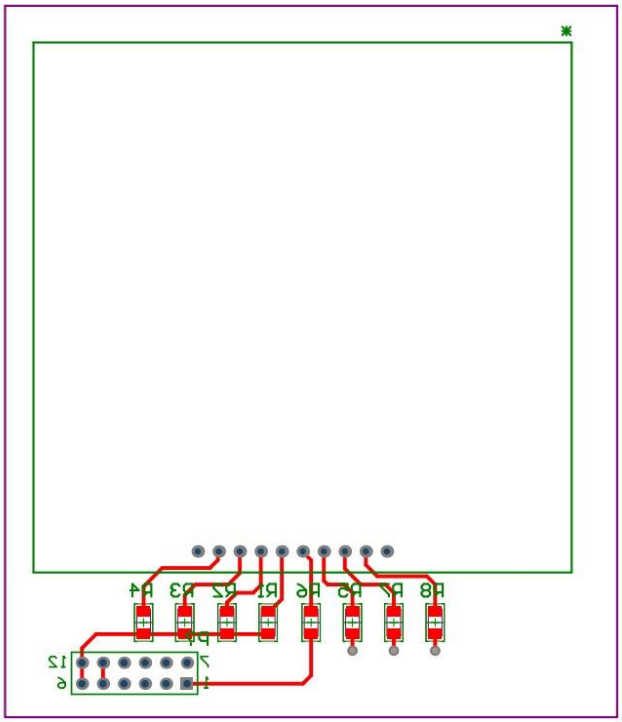
```
`timescale 1ns / 1ps
module DisplayController(
    DispVal,
    anode,
    segOut
    );
// ============================================================================================================
//                                          Port Declarations
// ============================================================================================================
    input [3:0] DispVal;            // Output from the Decoder
    output [3:0] anode;             // Controls the display digits
    output [6:0] segOut;            // Controls which digit to display
// ============================================================================================================
//                                  Parameters, Regsiters, and Wires
// ============================================================================================================
    // Output wires and registers
    wire [3:0] anode;
    reg [6:0] segOut;
// ============================================================================================================
//                                                  Implementation
// ============================================================================================================
    // only display the rightmost digit
    assign anode = 4'b1110;
    //------------------------------
    //         Segment Decoder
    // Determines cathode pattern
    //   to display digit on SSD
    //------------------------------
    always @(DispVal) begin
            case (DispVal)
                    4'h0 : segOut <= 7'b1000000;  // 0
                    4'h1 : segOut <= 7'b1111001;  // 1
                    4'h2 : segOut <= 7'b0100100;  // 2
                    4'h3 : segOut <= 7'b0110000;  // 3
                    4'h4 : segOut <= 7'b0011001;  // 4
                    4'h5 : segOut <= 7'b0010010;  // 5
                    4'h6 : segOut <= 7'b0000010;  // 6
                    4'h7 : segOut <= 7'b1111000;  // 7
                    4'h8 : segOut <= 7'b0000000;  // 8
                    4'h9 : segOut <= 7'b0010000;  // 9
                    4'hA : segOut <= 7'b0001000;    // A
                    4'hB : segOut <= 7'b0000011;    // B
                    4'hC : segOut <= 7'b1000110;    // C
                    4'hD : segOut <= 7'b0100001;    // D
                    4'hE : segOut <= 7'b0000110;    // E
                    4'hF : segOut <= 7'b0001110;    // F
                    default : segOut <= 7'b0111111;
            endcase
    end
endmodule
```

SCHEMATIC:

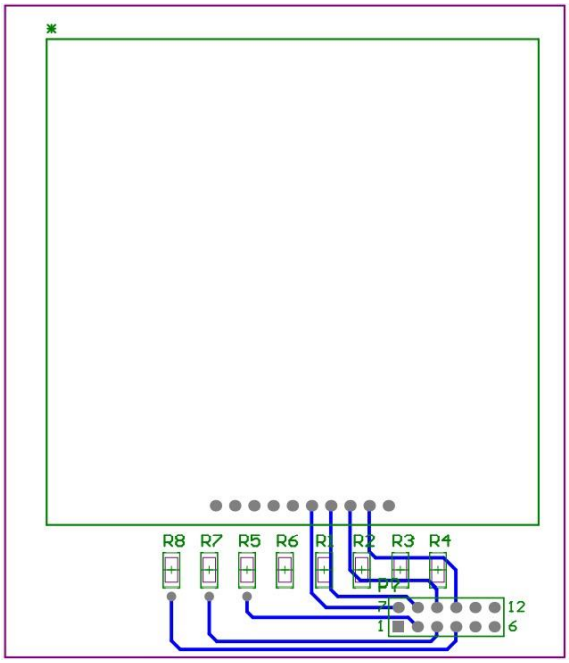


BOARD LAYOUT:

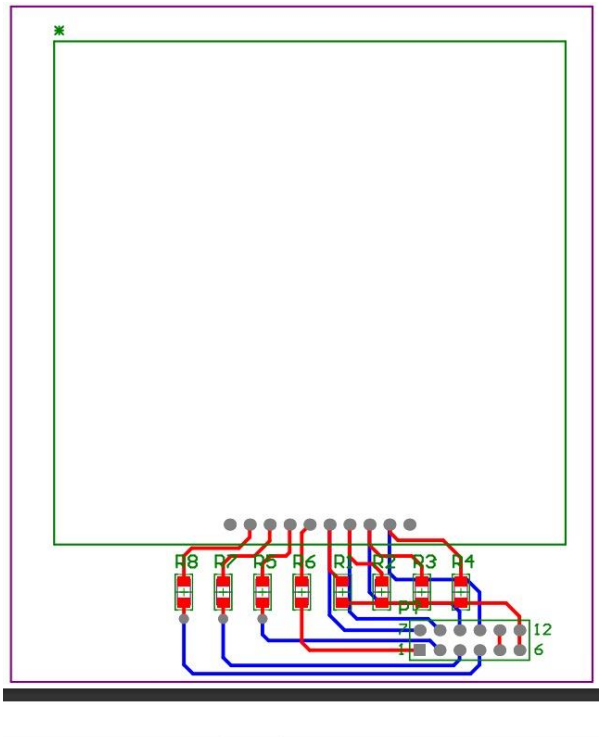
TOP:



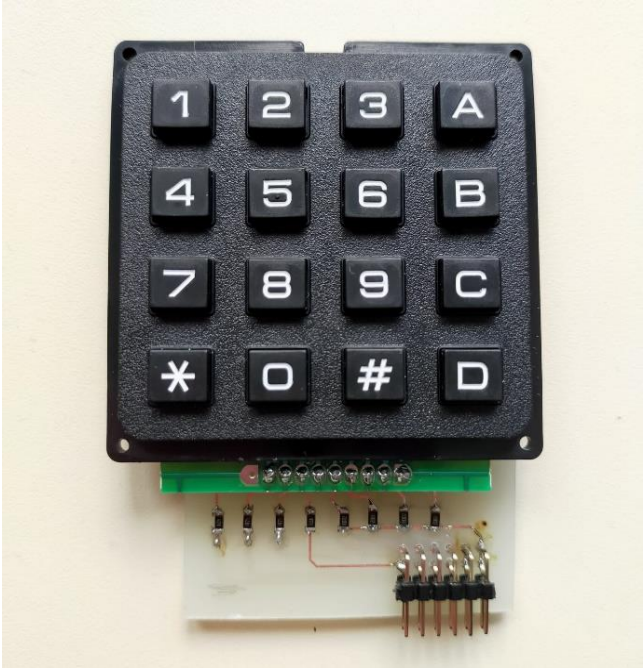
BOTTOM :

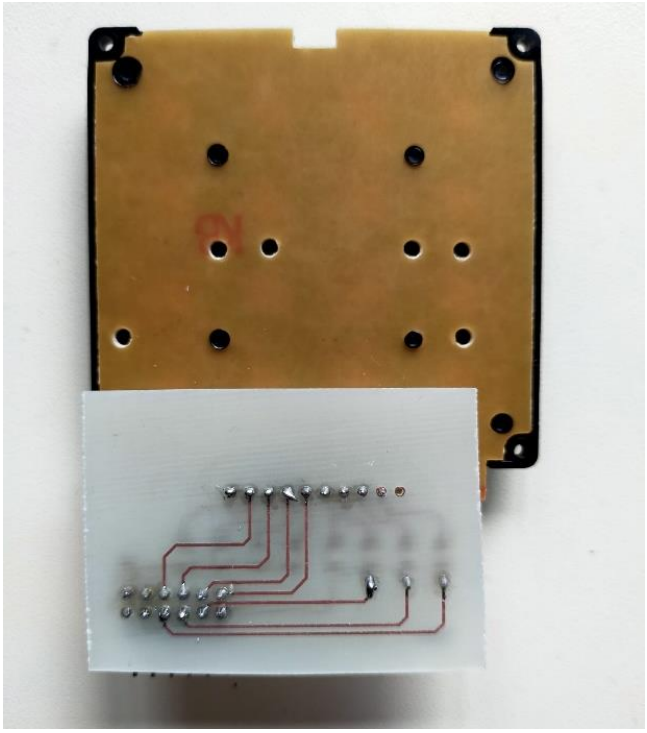


TOP/BOTTOM:



FOTAGE OF THE CIRCUIT





RESOURCES :

- https://digilent.com/shop/pmod-kypd-16-button-keypad/