```
`timescale 1ns / 1ps
////////////////////////////////////////////////////////////////////////////////
// Company:
// Engineer:  Titus Karuri
// Create Date: 10/19/2020 01:05:22 PM
// Module Name: main_project

module main_project(
    input clock,
    input sw15,
    input ips_1,
    input ips_2,
    input ips_3,
    input ips_4,
    input reset,
    input echo,
    output an1, // 4 digits on basys 3 board
    output an2,
    output an3,
    output an4,
    output LED0,
    output LED1,
    output LED2,
    output LED3,
    output LED15,
    output reg in1, // directional control pins
    output reg in2,
    output reg in3,
    output reg in4,
 output trig,
 output distance,
 output segment0,
 output segment1,
 output segment2,
 output segment3,
 output segment4,
 output segment5,
 output segment6,
 output enable_a,
 output enable_b,
 input comparator_1,
 input comparator_2

 );

reg[3:0] pwm_temp;
reg[7:0] segment_temp; // 7 bit register to hold input
reg[3:0] an_temp;
reg[30:0] stop;
reg[18:0] count;
reg[20:0] widthL;
reg[20:0] widthR;
reg[20:0] width;
reg[20:0] counter;
reg[3:0] timer;

    initial begin
     counter =0;
```

```
stop=0;
count =0;
//speed = 0;
width = 0;
pwm_temp=0;

end

assign enable_a = pwm_temp;
assign enable_b = pwm_temp;

   always @(*) // sets different switches and segments to assigned roles
  begin
    an_temp[0] <= an1;
    an_temp[1] <= an2;
    an_temp[2] <= an3;
    an_temp[3] <= an4;
    segment_temp[0] <= segment0;
    segment_temp[1] <= segment1;
    segment_temp[2] <= segment2;
    segment_temp[3] <= segment3;
    segment_temp[4] <= segment4;
    segment_temp[5] <= segment5;
    segment_temp[6] <= segment6;



end
```

```
always @(posedge clock)begin //pwm signal

      if(stop==0)
       begin
        if(comparator_1==1||comparator_2==1)
       begin
        counter <= counter +1;
       end
         if (counter == 3000000) // 60 hertz
        begin
         stop <= 1;
        end
         count <= count+1;

         if(count < width)
         begin
             // creates pwm
           pwm_temp = 1;

          end
           else
          begin

           pwm_temp = 0;
           widthL = 0;
          end
          end
           else
          begin
         if (stop == 1)  //  comparator sends signal to stop
        begin
          stop <= 1;
        end
         stop <= stop + 1;  // shuts dowm h bridge and rover
           // shuts down pwm
         pwm_temp = 0;
         widthL = 0;

         if (reset) // reset button
         begin
         stop <= 0;  // restarts counter
         end

        end
        end




always @(posedge clock) begin

     if(ips_1==1 & ips_2==1 & ips_3==1 & ips_4==1)
       begin
        in1=0;
        in2=0;
        in3=0;
        in4=0;
       end
```

```
  else if(ips_1 == 0)
  begin

  width =20'd50000000;

  in1  = 1;
  in2  = 0;
  in3  = 1;
  in4  = 0;
end

else if(ips_2 == 0)
  begin

  width =20'd2368421;
   in1  = 0;
   in2  = 1;
   in3  = 1;
   in4  = 0;
  end

else if(ips_3 == 0)
  begin

  width =20'd2368421;
   in1  = 0;
   in2  = 1;
   in3  = 1;
```

```
         in4  = 0;
       end

      else if(ips_4 == 0)
        begin

      width =20'd50000000;
         in1  = 0;
         in2  = 1;
         in3  = 0;
         in4  = 1;
      end

    end




assign LED0 = ~ips_1;
assign LED1 = ~ips_2;
assign LED2 = ~ips_3;
assign LED3 = ~ips_4;
assign LED15 = sw15;
assign an1 = an_temp[0];
assign an2 = an_temp[1];
assign an3 = an_temp[2];
```

```
  assign an4 = an_temp[3];
  assign segment0 = segment_temp[0];
  assign segment1 = segment_temp[1];
  assign segment2 = segment_temp[2];
  assign segment3 = segment_temp[3];
  assign segment4 = segment_temp[4];
  assign segment5 = segment_temp[5];
  assign segment6 = segment_temp[6];

endmodule
```