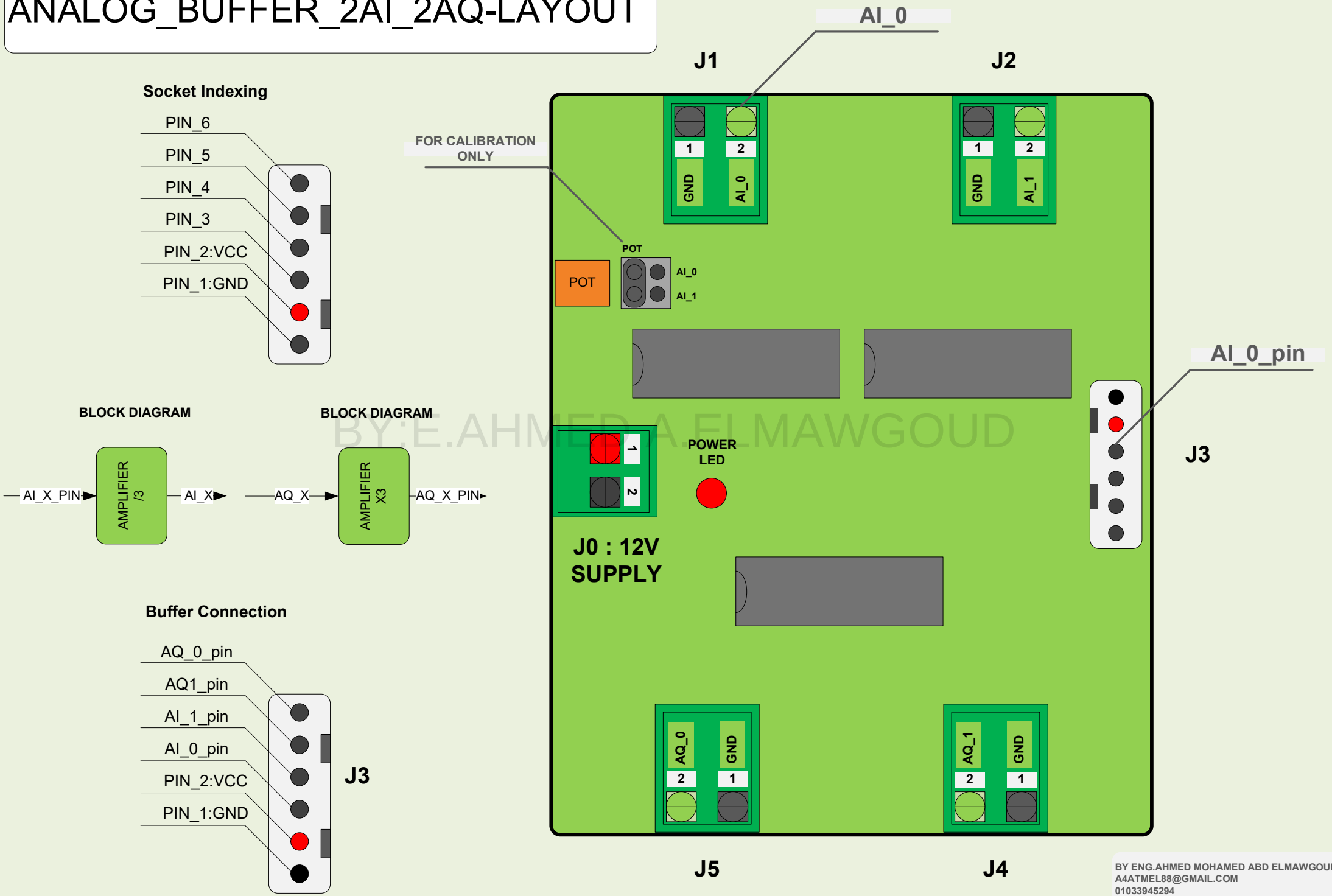
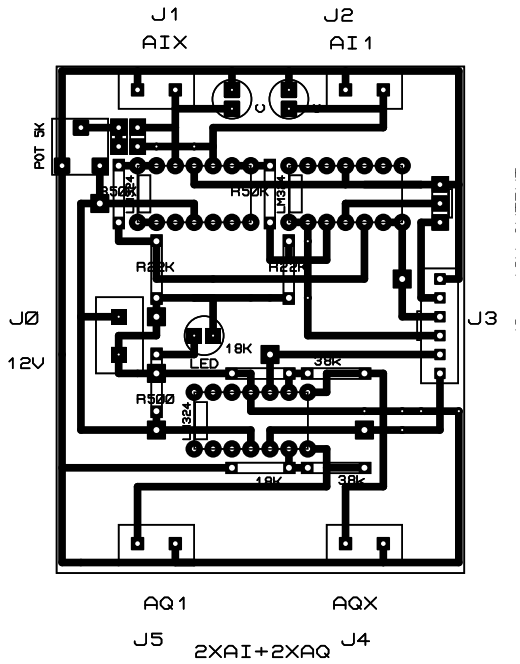
ANALOG_BUFFER_2AI_2AQ-LAYOUT





```
/*
 * DESIGNED BY ENG.AHMED MOHAMED ABD ELMAWGOUD
 * A4ATMEL88@GMAIL.COM
 * USE THE IO NAMING ACOORDIG TO drawing
ANALOG_BUFFER_2AI_2AQ-LAYOUT.PDF
 * FOR ANALOG OUTPUT ,IT PRODUCE OUTPUT VOLTAGE FORM 0  TO 8 VOLT
BY PASSING  VALUES FROM 0 TO 80
 * FOR ANALOG INPUT , IT READ THE INPUT VOLTAGE FROM O TO 8 VOLT ,
AND RETRURN VALUES FROM 0 TO 80
*/

#include <ESP32_BOARD.h>
#include <ANALOG_BUFFER_2AI_2AQ.h>

int ai_0_value ,ai_1_value;

void setup() {
  Serial.begin(115200);
}

void loop() {
  // AI0
  ai_0_value = read_ai0();                //return value from 0 to
80 [voltage from 0 to 8 v ,values from 0 to 80]
  Serial.print("AI_0 value = ");
  Serial.println( ai_0_value );
  // AI1
  ai_1_value = read_ai1();                //return value from 0 to
80 [voltage from 0 to 8 v ,values from 0 to 80]
  Serial.print("AI_1 value = ");
  Serial.println( ai_1_value );
  // AQ0
  write_aq0( 13 );   // produce 1.3 v on AQ_0_pin [voltage from 0
to 8 v ,values from 0 to 80]
  // AQ1
  write_aq0( 54 );   // produce 5.4 v on AQ_0_pin [voltage from 0
```

```
to 8 v ,values from 0 to 80]
}

/*



// ANALOG_BUFFER_2AI_2AQ IS CONNECTED TO CONTROLLER J5 SOCKET
/* --
 * DESIGNED BY ENG.AHMED MOHAMED ABD ELMAWGOUD
 * A4ATMEL88@GMAIL.COM
 * functions
 *  int read_ai0()          -- read from 0 to 100 [ 0v to 10v]
 *  int read_ai1()
 *  write_aq0(int value)    -- place from  0v to 10v [numbers from
0 to 100]
 *  write_aq1(int value)
 */

////////////////////////////////////////// ONLY REPLACE J5 TO
THE CONNECTED SOCKET
#define AI_0_pin  j5_pin3
#define AI_1_pin  j5_pin4
#define AQ_1_pin  j5_pin5
#define AQ_0_pin  j5_pin6

////////////////////////////////////////////// int map_analog(int
adc_value)
int map_analog(int adc_value)
{
  int int_volt;
  if (adc_value < 140) int_volt =       ( adc_value * 10 ) / 140;
// int_volt <= 1 volt
  else if (adc_value < 425) int_volt =  ( adc_value * 20 ) / 425;
// int_volt <= 2 volt
  else if (adc_value < 720) int_volt =  ( adc_value * 30 ) / 720;
// int_volt <= 3 volt
  else if (adc_value < 990) int_volt =  ( adc_value * 40 ) / 990;
```

```
// int_volt <= 4 volt
  else if (adc_value < 1285) int_volt = ( adc_value * 50 ) / 1285;
// int_volt <= 5 volt
  else if (adc_value < 1580) int_volt = ( adc_value * 60 ) / 1580;
// int_volt <= 6 volt
  else if (adc_value < 1855) int_volt = ( adc_value * 70 ) / 1855;
// int_volt <= 7 volt
  else if (adc_value < 2145) int_volt = ( adc_value * 80 ) / 2145;
// int_volt <= 8 volt
  else if (adc_value < 2300) int_volt = ( adc_value * 85 ) / 2300;
// int_volt <= 8.5 volt
  else if (adc_value > 2300) int_volt = 86; // int_volt 0= 8.5 volt
  return int_volt;
}
////////////////////////////////////////////////////////////--
int read_ai1()
int read_ai0()  // return value from 0 to 100 as innput ranges from
0 to 10v
{
  int adc_value = analogRead(AI_0_pin);
  int voltage = map_analog(adc_value);
  return voltage;
}
////////////////////////////////////////////////////////////--
int read_ai1()
int read_ai1()  // return value from 0 to 100 as innput ranges from
0 to 10v
{
  int adc_value = analogRead(AI_1_pin);
  int voltage = map_analog(adc_value);
  return voltage;
}
////////////////////////////////////////////////////////////--
void write_ao1(int value)
void write_aq0(int value)  // value from 0 to 100 make output from
0 to 10 volts
{
```

```
  int dac_val = (value*512)/200;
  dacWrite(AQ_1_pin ,dac_val);
}
////////////////////////////////////////////////////////////////////--
void write_ao1(int value)
void write_aq1(int value)  // value from 0 to 100 make output from
0 to 10 volts
{
  int dac_val = (value*512)/200;
  dacWrite(AQ_1_pin ,dac_val);
}


*/
```