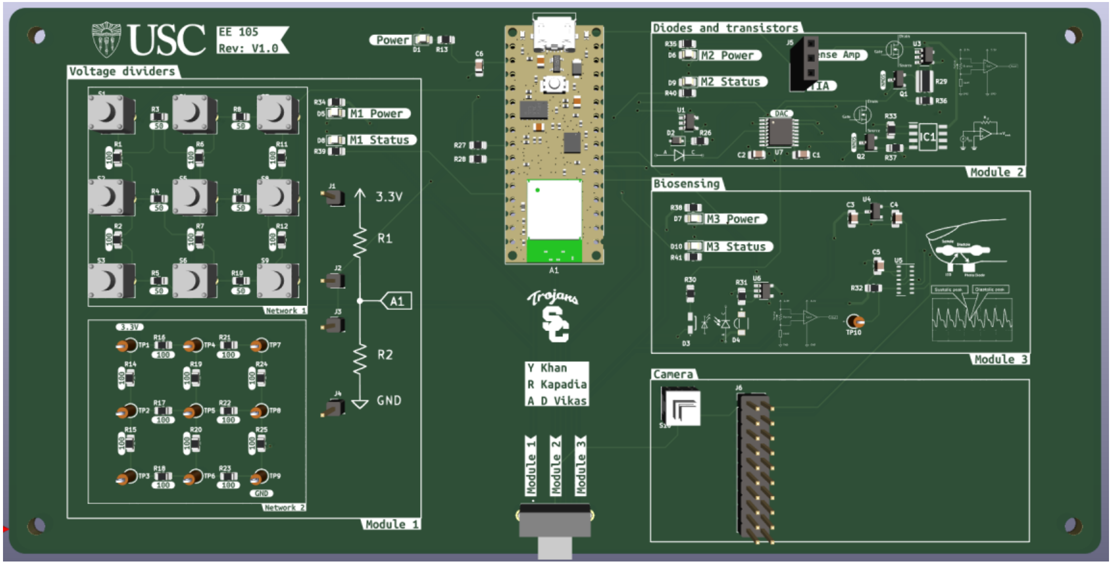
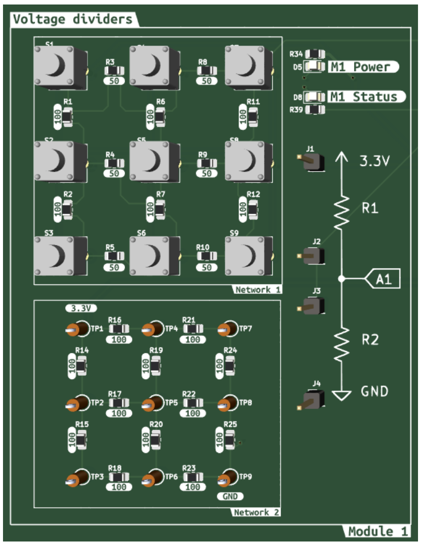
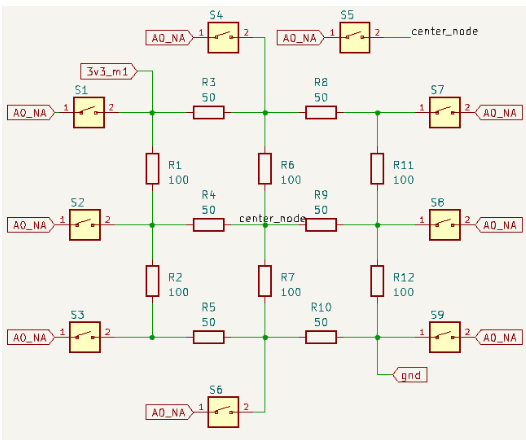


1. **Module 1:**



   a. **Uneven network array:**
   A resistor grid was built in a checkerboard pattern where all the horizontal resistors are 50 ohms and all the vertical resistors are 100 ohms. The top left corner of the grid is connected to a 3.3V power source, and the bottom right corner is connected to the ground. Each node in the grid is connected to an analog input pin on an Arduino through a push button.



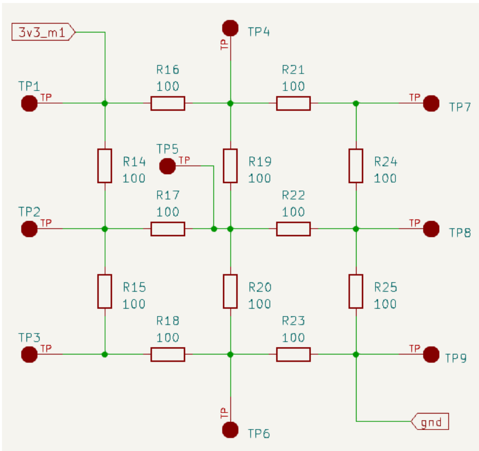
Uneven network array schematic

In the above setup, 'A0_NA' is the analog input pin on the Arduino. This setup can be used as a number pad. When a push button connected to a particular node is pressed, it produces an analog value between 0V and 3.3V. Because the resistor network is uneven, each node has a different analog value, making it easy to differentiate and identify each node by its analog value. Below is sample code that shows the index of the button pressed.

```
#define m1_na A0
void m1_network_array(){
  int analog_value = analogRead(m1_na);
  if (analog_value > 1000 && analog_value < 1020) Serial.println("S1");
  else if (analog_value > 285 && analog_value < 305)Serial.println("S2");
  else if (analog_value > 185 && analog_value < 210)Serial.println("S3");
  else if (analog_value > 690 && analog_value < 715)Serial.println("S4");
  else if (analog_value > 335 && analog_value < 360)Serial.println("S5");
  else if (analog_value > 140 && analog_value < 160)Serial.println("S6");
  else if (analog_value > 560 && analog_value < 580)Serial.println("S7");
  else if (analog_value > 305 && analog_value < 325)Serial.println("S8");
  else if (analog_value > 60 && analog_value < 100)Serial.println("S9");
}
```

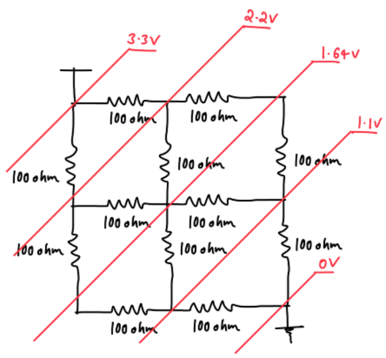
b. Even network array:

In this design, the resistors are arranged in a checkerboard pattern, but unlike the uneven network array, all the resistors have the same value. As a result, all the nodes on the diagonal have the same potential. Below is the schematic of this even network array.



Even network array schematic

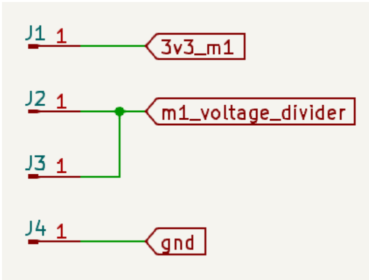
Test points are available at every node to prob and measure the node voltage. The left top node is connected to the 3.3V voltage source and the right bottom node is connected to the GND ground.



Node voltages in even network array

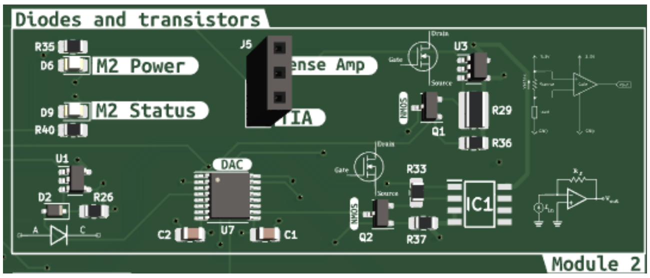
c. Voltage Divider:

Two placeholders are connected in series between 3.3V and GND to hold through-hole package resistors. The center point of these resistors is connected to the analog input of the Arduino to measure the voltage at the center node. Below is the schematic. The 'm1_voltage_divider' is the analog input pin of the Arduino.



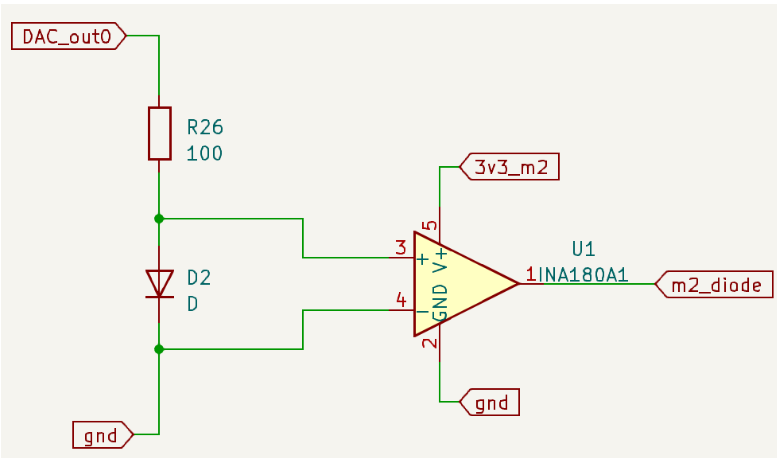
Voltage divider schematic

2. Module 2:



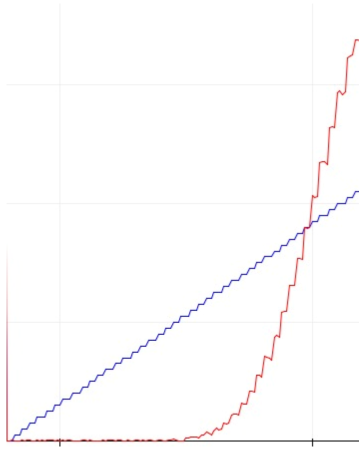
a. Diode:

Diode is connected in series with a resistor and powered by a DAC (Digital to Analog Converter) to control the voltage across the resistor-diode pair. The voltage across the resistor is measured using an Op-Amp to amplify the voltage drop across the resistor, which is then sent to the analog input of the Arduino. The voltage drop across the diode is proportional to the current through the diode. Below is the schematic of the discussed setup.

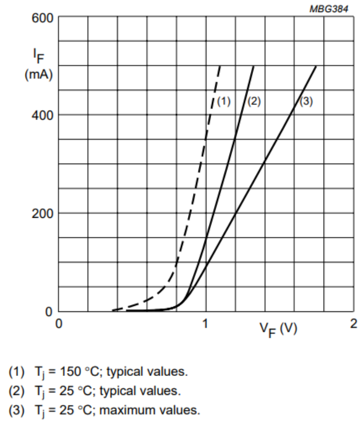


Diode current measurement setup schematic

The DAC used in this design has eight channels. Channel zero is powering the diode section of the dev-board. ‘DAC_out0’ is the output from channel zero of the DAC. ‘m2_diode’ is the analog input of the Arduino. In the graph below, the blue curve represents the DAC output, and the red curve represents the current through the diode.



measured forward current and forward voltage of the diode



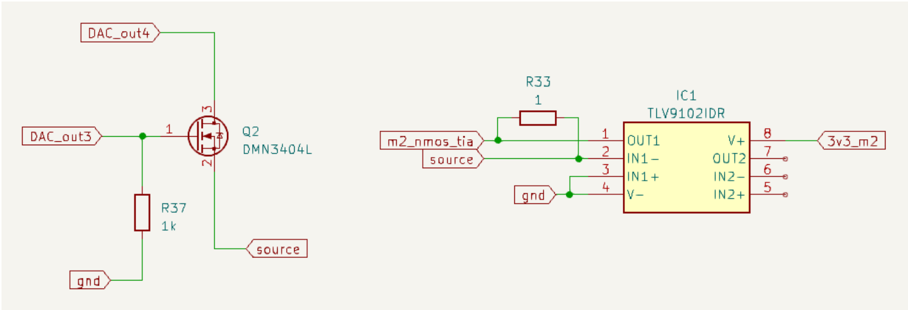
forward current and forward voltage of the diode from the data sheet

Below is the sample code that records the current through the diode using the above discussed setup

```
#define dac_diode 0
#define m2_diode A2

void diode(){
  int inc_interval = 15;
  int inc_delay = 1;
  int inc = 0;
  unsigned long prev_inc_time = 0;
  while(1){
    Serial.print(inc);
    Serial.print(", ");
    Serial.println(analogRead(m2_diode));
    if((inc <= 650) && (millis() - prev_inc_time > inc_delay)){
      prev_inc_time = millis();
      setDACVal(dac_diode, inc);
      inc = inc + inc_interval;
    }
    if((inc >= 650) && (millis() - prev_inc_time > inc_delay)){
       prev_inc_time = millis();
       setDACVal(dac_diode, 0);
       inc = 0;
    }
  }
}
```

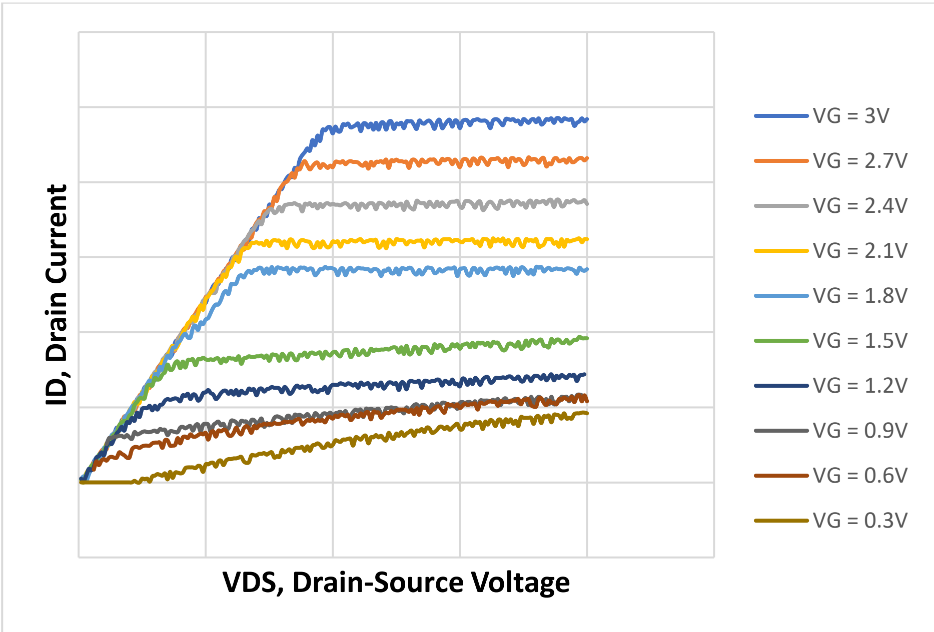
b. NMOS:

The drain and gate voltages of the NMOS are controlled via a DAC. The drain voltage is controlled using channel 4 of the DAC, and the gate voltage is controlled using channel 3 of the DAC. The source of the NMOS is connected to a TIA (Transimpedance Amplifier) to measure the channel current (Drain Current Id). Below is the schematic of the setup discussed above.

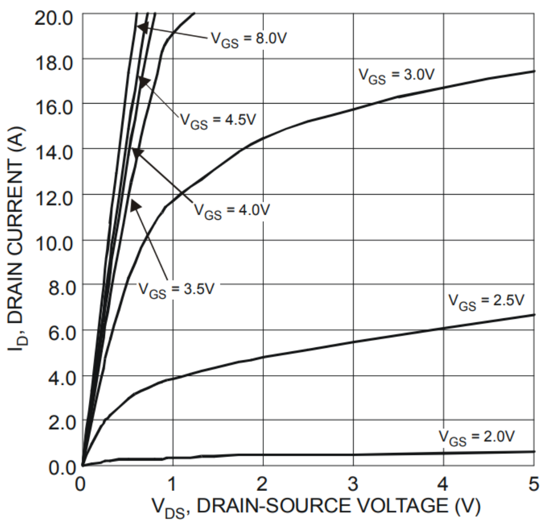


NMOS channel current measurement schematic

'TLV9102IDR' is dual Op-Amp IC. It is used as TIA in the above design. The amplification of the current is set to '1' buy placing a 1ohm resistor as the feedback resistor. Below shown are the results measured using the above setup.



Drain current measures using TIA

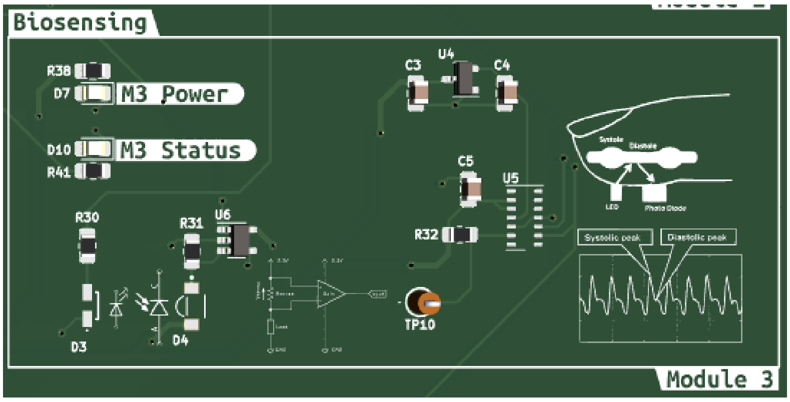


Drain current vs Drain-Source Voltage graph from the data sheet

Below shown is the sample code that produces the above results with the above discussed set-up

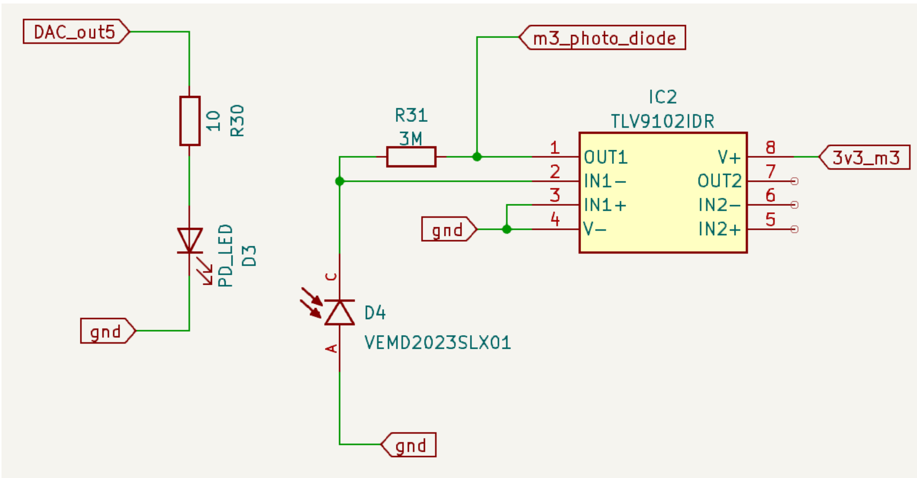
```
#define dac_nmos2_gate 3
#define dac_nmos2_drain 4
#define m2_nmos A3

void nmos_tia(){
  for(int k = 0; k < 2000; k = k + 200){
    setDACVal(dac_nmos2_gate, k);
    for(int i = 0; i < 2000; i = i + 10){
      setDACVal(dac_nmos2_drain, i);
      Serial.println(analogRead(m2_nmos));
      delay(10);
    }
  }
}
```

3. Module 3:



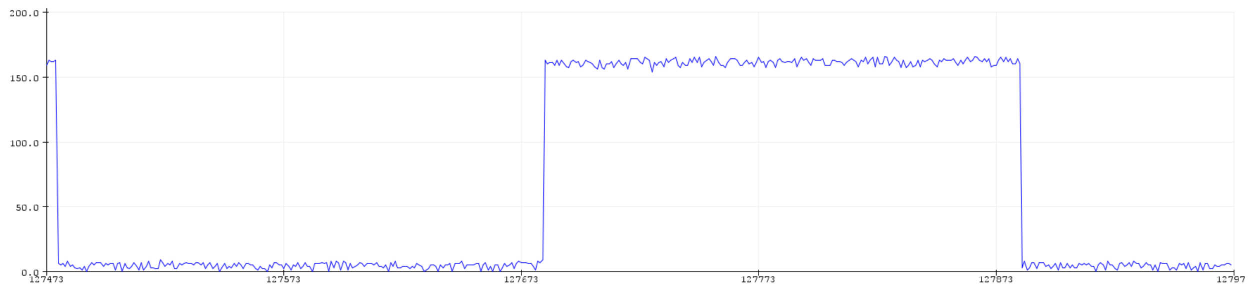
a. Photo Diode:

An LED with a resistor is connected to the DAC over channel 5. The photodiode is connected to the TIA (Transimpedance Amplifier) to record the current. A large feedback resistance of 3M ohms is used to amplify the tiny currents produced by the photodiode. Below is the schematic of the setup discussed above. Blow image shoes the schematic of the setup



LED and photo diode setup schematic

As discussed before, a ‘TLV9102IDR’ is used as the TIA (Transimpedance Amplifier). The output of this TIA is connected to the analog input of the Arduino, which is labeled as ‘m3_photo_diode’. Below is the output of the TIA when the LED is flashed at a particular frequency. Using this setup, a message can be encoded using frequency modulation techniques and decoded using the recorded pulses from the TIA output.



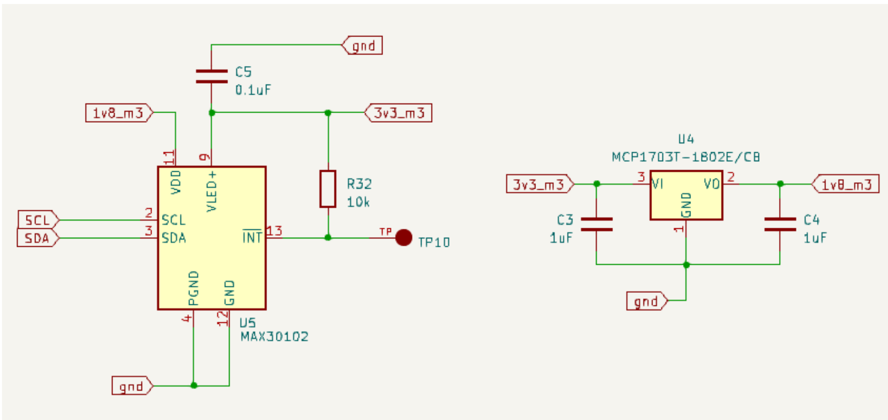
Amplified photo diode current when an LED is flashed on it

Below shown is the sample code to record the above shown result using the above setup.

```
#define m3_pd A6
#define dac_pd 5

void flash(){
    int time_period = 100;
    int previous_state = 0;

    unsigned long previous_toggle_time = 0;


    while(1){
      Serial.println(analogRead(m3_pd));
      if(millis() - previous_toggle_time >= time_period/2){
        previous_toggle_time = millis();
        if(previous_state == 0){
          setDACVal(dac_pd, 2250);
          previous_state = 1;
        }
        else{
          setDACVal(dac_pd, 0);
          previous_state = 0;
        }
      }
    }
}
```

b. PPG:

MAX30102 a AFE (Analog Fron End) from the Analog Devices is used to measure the PPG of the user. This AFE communicates with the Arduino over I2C. Blow shown is the schematic of the setup.



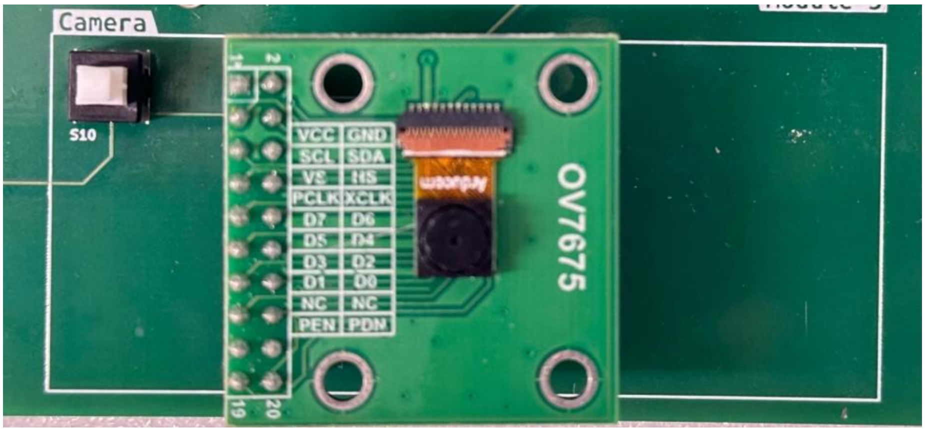
MAX30102 schematic

Below shown is the PPG (Heart rate and Spo2) measured using the above setup

```
04:07:51.622 -> Heart rate:26.84bpm / SpO2:94%
04:07:52.594 -> Beat!
04:07:52.635 -> Heart rate:39.82bpm / SpO2:94%
04:07:53.645 -> Heart rate:39.82bpm / SpO2:94%
04:07:54.250 -> Beat!
04:07:54.654 -> Heart rate:37.79bpm / SpO2:94%
04:07:54.816 -> Beat!
04:07:55.617 -> Heart rate:60.26bpm / SpO2:94%
04:07:56.626 -> Heart rate:60.26bpm / SpO2:94%
04:07:57.518 -> Beat!
```

Heart rate and Spo2 measured using MAX30102

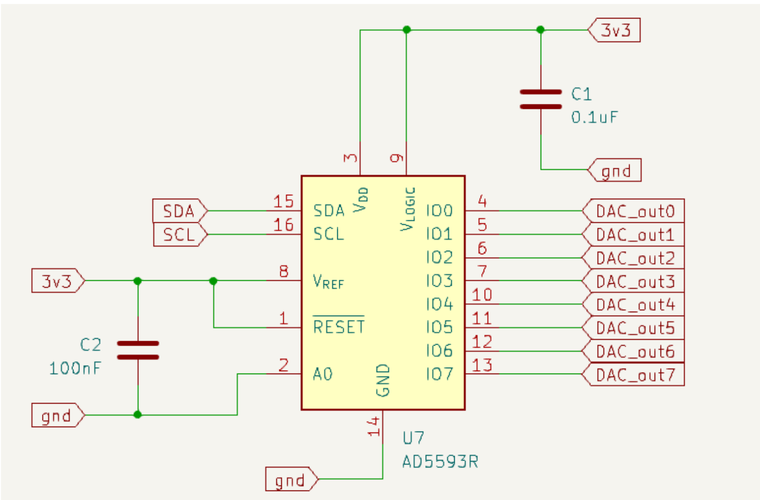
4. **Camera:**
   OV7675 camera module work on I2C which can be powered using a push lock switch 'S10' as shown in the below image.



Camera module and on/off push lock switch

5. **Additional Circuitry:**

   This dev-board has some additional circuits, including a DAC used in various modules of the design, and some indicator LEDs. Below is the schematic of the DAC.



AD5593R DAC Schematic