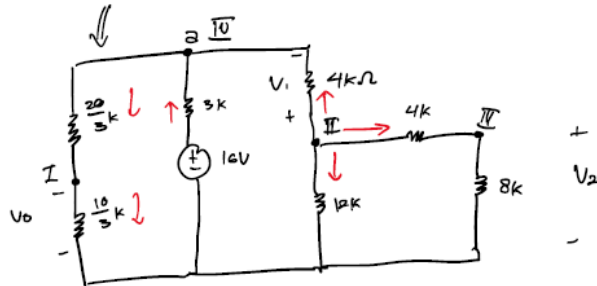
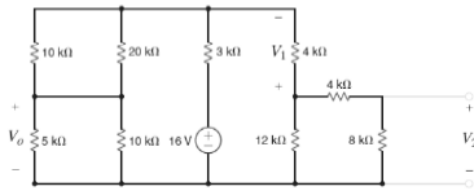


Electrical Test

17)



Analysis Node

$$\text{Node I} \quad \frac{V_0 - V_0}{20/3k} = \frac{V_0}{10/3k}$$

$$\text{Node II} \quad \frac{V_1 - V_0}{4k} + \frac{V_1}{12k} + \frac{V_1 - V_2}{4k} = 0$$

$$\text{Node III} \quad \frac{V_1 - V_2}{4k} = \frac{V_2}{8k}$$

$$\text{Node IV} \quad \frac{16 - V_0}{3k} + \frac{V_1 - V_0}{4k} = \frac{V_0 - V_0}{20/3k}$$

ELIMINASI

$$\begin{aligned} \frac{V_0 - V_0}{20/3k} &= \frac{V_0}{10/3k} \\ \frac{1}{20} (V_0 - V_0) &= \frac{3}{10} (V_0) \\ \frac{1}{20} (V_0 - V_0) &= \frac{1}{10} V_0 \\ V_0 - V_0 &= 2V_0 \\ V_0 &= 3V_0 \end{aligned}$$

$$\begin{aligned} \frac{V_1 - 3V_0}{4k} + \frac{V_1}{12k} + \frac{V_1 - V_0}{4k} &= 0 \\ 3(V_1 - 3V_0) + V_1 + 3(V_1 - V_0) &= 0 \\ 3V_1 - 9V_0 + V_1 + 3V_1 - 3V_0 &= 0 \\ V_1 - 3V_0 - 4V_0 &= 0 \quad (1) \end{aligned}$$

$$\begin{aligned} \frac{V_1 - V_2}{4k} &= \frac{V_2}{8k} \\ 2V_1 - 2V_2 &= V_2 \\ 2V_1 &= 3V_2 \\ V_1 &= \frac{3}{2} V_2 \\ V_1 - \frac{3}{2} V_2 &= 0 \quad (2) \end{aligned}$$

$$\begin{aligned} \frac{16 - 3V_0}{3k} + \frac{V_1 - 3V_0}{4k} &= \frac{3}{20} V_0 - V_0 \\ \frac{16 - 3V_0}{3} + \frac{V_1 - 3V_0}{4} &= \frac{3}{20} 3V_0 - V_0 \\ \frac{16 - 3V_0}{3} + \frac{V_1 - 3V_0}{4} &= \frac{3}{10} V_0 \end{aligned}$$

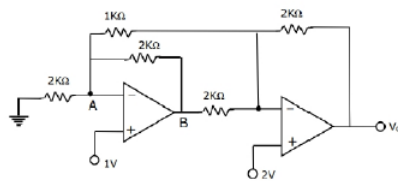
$$\begin{aligned} \frac{16 - 3V_0}{3} + \frac{V_1 - 3V_0}{4} &= \frac{3}{10} V_0 \\ 40(16 - 3V_0) + 30(V_1 - 3V_0) &= 36V_0 \\ 640 - 120V_0 + 30V_0 - 90V_0 &= 36V_0 \\ 640 - 180V_0 &= 36V_0 \\ 640 &= 216V_0 \\ V_0 &= 2.96V \end{aligned}$$

$$\begin{pmatrix} -4 & 7 & -3 \\ 0 & 1 & -3/2 \\ 246 & -30 & 0 \end{pmatrix} \begin{pmatrix} V_0 \\ V_1 \\ V_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 640 \end{pmatrix}$$

dioperasikan
Menggunakan MATLAB

$$\begin{aligned} V_0 &= 3,33V \\ V_1 &= 6V \\ V_2 &= 4V \end{aligned}$$

2)

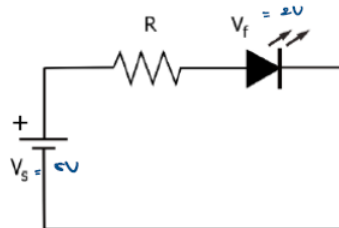


$$V_0 = \left(1 + \frac{2k}{2k}\right)1 + \left(1 + \frac{2k}{2k}\right)2$$

$$= (2) \cdot 1 + (1+1)2$$

$$= 2 + 4 = \underline{\underline{6V}}$$

3



a) KVL : $V_f - V_S - V_R = 0$

$$V_R = V_S - V_f$$

$$V_R = 5 - 2 = 3V$$

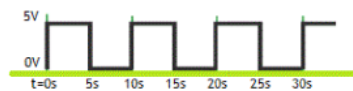
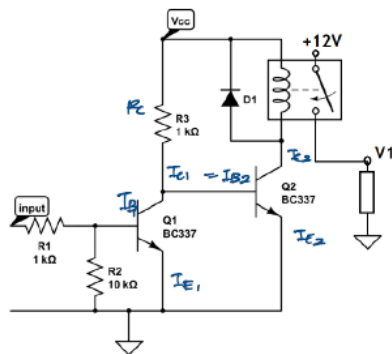
$$R = \frac{V_R}{I_R} = \frac{3V}{15 \cdot 10^{-3}A} = \underline{\underline{200\Omega}}$$

b) $P_{LED} = V_f \cdot I$

$$= 2V \cdot 15mA$$

$$= \underline{\underline{30mW}}$$

4)



$$V_{input} = 5V$$

$$I_B = \frac{5V - 1.2}{\frac{10}{11}k}$$

$$= \frac{3.8 \cdot 11}{\frac{10 \cdot 1000}{10000}} = 4.18mA \rightarrow \text{BJT Q1 ON}$$

$$\frac{1}{R_1 \parallel R_2} = \frac{1}{10} + \frac{1}{1} = \frac{11}{10}$$

$$R_1 \parallel R_2 = \frac{10}{11}$$

BJT Q1 ON

$$I_{E1} = I_{C1} + I_{B1}$$

$$= 4.18 + 5$$

$$= 9.18$$

$$I_{C1} = \frac{V_{CC}}{R_C} = \frac{5}{1000} = 5mA$$

BJT Q1 akan ON, maka $I_{C1} = I_{B2} \approx 0$, yang membuat BJT Q2 OFF.

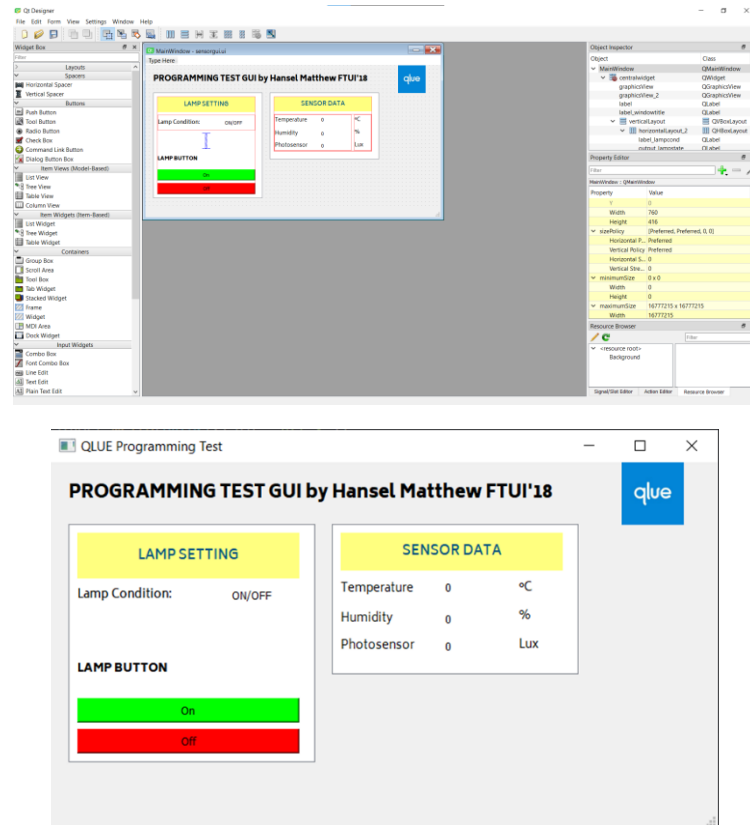
Karena BJT Q2 OFF, maka arus seluruhnya akan mengalir dari VCC ke arah BJT Q1. Sehingga beda potensial pada coil relay akan 0 yang membuat relay open

Karena relay open, maka tegangan V1 akan 0 karena rangkaian disebelah kanan akan open

Programming Test

#MERANCANG GUI

Tahapan awal adalah membuat rancangan GUI di QT Designer, untuk membuat sebuah GUI dapat menggunakan QT atau Tkinter namun saya menggunakan QT untuk simplifitas dan menghemat waktu. Pada QT Designer dibuat tombol – tombol, label, judul dan aset GUI lainnya. File GUI disimpan dalam file **sensorgui.ui**. Berikut merupakan gambar GUI yang didesain pada QT Designer.



Pada GUI terdapat 3 buah komponen utama yaitu:

- Lamp condition yang akan memberitahu kondisi lampu
- Tombol ON/OFF untuk menyalakan atau mematikan lampu
- Ketiga buah label untuk menampilkan bacaan dari sensor.

#KODE PYTHON

Selanjutnya adalah membuat kode python dari GUI tersebut. Kode tugas ini saya buat di file **sensorcode.py**. Kode dimulai dengan memasukkan library yang akan digunakan pada kode sumber

```
#Gui Library
from PyQt5.QtWidgets import *
from PyQt5.uic import loadUi
from PyQt5 import QtGui

#System library
import sys

#Import logo resource file
import logo_rc

#Serial library for communicating to microcontroller (arduino)
import serial
```

Kemudian saya membuat variabel serialPort yang akan bertanggung jawab atas komunikasi antara kode python dengan mikrokontroler.

```
#Declare serial port and baud rate (replace as needed but must be the same value as the value in the microcontroller)
serialPort = serial.Serial(port = "COM3", baudrate=9600, bytesize=8, timeout=2, stopbits=serial.STOPBITS_ONE)
serialString = "" # Used to hold data coming over UART
```

Selanjutnya saya membuat class dari GUI dan pada constructor class tersebut mensetting agar class tersebut terkoneksi dengan file **sensorgui.ui** yang sudah dibuat. Pada constructor juga dihubungkan method yang akan dijalankan ketika tombol pada gui ditekan

```
#Make GUI class
class sensor_gui (QMainWindow) :
    def __init__(self):
        #Load qt & gui
        QMainWindow.__init__(self)
        loadUi("sensorgui.ui",self)

        #Gui title & logo
        self.setWindowTitle("QLUE Programming Test")

        #Display gui
        self.show()

        #Button method
        self.TombolLampOn.clicked.connect(self.lampOn_pressed)
        self.TombolLampOff.clicked.connect(self.lampOff_pressed)
```

Selanjutnya saya membuat dua buah method yakni satu untuk tombol ON dan satu untuk tombol OFF. Ketika tombol ON ditekan, maka kode python akan mendisplay tulisan "ON" pada GUI dan mengirimkan pesan state on ke mikrokontroler. Ketika tombol off ditekan, GUI akan mendisplay tulisan "OFF" dan mengirimkan pesan state off juga ke mikrokontroler

```

#Button ON Pressed Function
def lampOn_pressed(self):
    #Display ON di GUI
    self.output_lampstate.setText("ON")
    state = 1
    serialPort.write(bytes(state, 'utf-8'))

#Button OFF Pressed Function
def lampOff_pressed(self):
    #Display OFF di GUI
    self.output_lampstate.setText("OFF")
    state = 0
    serialPort.write(bytes(state, 'utf-8'))

```

Selanjutnya adalah membuat objek dari QT dan objek dari class yang telah dibuat sebelumnya.

```

#Make GUI Object
app = QApplication([])
mainwindow = sensor_gui()
app.setQuitOnLastWindowClosed(True)

```

Setelah itu, bagian terakhir merupakan kode untuk membaca pesan dari mikrokontroller, dimana pesan tersebut dikirimkan dalam sebuah string dan dipisahkan dengan sebuah spasi. Ketika pesan diterima, pesan akan dipisah menjadi tiga buah variabel yang terpisah yaitu photosensor, humidity dan temperature, kemudian ketiga nilai tersebut akan didisplay di GUI menggunakan method .setText()

```

# #Sensor reading
while 1:
    # Wait until there is data waiting in the serial buffer
    if serialPort.in_waiting > 0:

        # Read data out of the buffer until a carriage return / new line is found
        serialString = serialPort.readline()

        # Display the data to the GUI
        try:
            #Split the incoming message to multiple data
            serialString = serialString.decode("Ascii").split()

            #Save the value into separate variable
            photosensor = float(serialString[0])
            humidity = float(serialString[1])
            temperature = float(serialString[2])

            #Display the value on the GUI
            mainwindow.output_photosensor.setText(str(round(photosensor ,3)))
            mainwindow.output_humidity.setText(str(round(humidity ,3)))
            mainwindow.output_temperature.setText(str(round(temperature ,3)))

        except:
            pass

```

Saya juga membuat file **guionly.py** dimana saya sudah comment kode yang berhubungan dengan komunikasi mikrokontroller jika ingin melihat tampilan gui tanpa terkoneksi atau mengatur mikrokontroller terlebih dahulu.

#KODE ARDUINO

Selanjutnya saya membuat kode pada mikrokontroler yaitu **sensor_arduino.ino** yang tersimpan pada folder **sensor_arduino**. Kode dimulai dengan memasukkan library dan mendefine variabel yang akan digunakan. Saya mengambil asumsi bahwa sensor temperatur dan humidity yang digunakan merupakan **sensor DHT11**.

```
#include "DHT.h" //Temp and Humidity Sensor Library

#define DHTPIN 2    // DHT Pin number

#define DHTTYPE DHT11 // DHT sensor type, i use DHT11 sensor
```

Selanjutnya saya mendeklarasikan beberapa variabel yang akan digunakan pada kode tersebut.

```
float ADC_value = 0.0048828125; // ADC resolution
int lampPin = 11;

//Message variable
char msg[60];
char outstr_photosensor[15];
char outstr_humid[15];
char outstr_temp[15];
```

Pada bagian void setup, dilakukan pemulaian komunikasi serial dan dht serta mengatur jenis pin 11 menjadi output

```
void setup() {
  Serial.begin(9600); //begin serial communication
  dht.begin(); //begin dht communication
  pinMode(lampPin,OUTPUT);
}
```

Pada bagian awal void loop, terdapat delay untuk reading interval, dan juga kode untuk membaca pesan dari python via serial. Ketika pesan diterima, jika pesan yang diterima bernilai 1, maka mikrokontroler akan menyalakan lampu, dan jika pesan yang diterima bernilai 0, maka mikrokontroler akan mematikan lampu

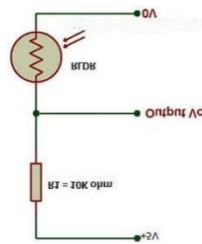
```
void loop() {
  delay(1000); //Sensor Reading Interval 1Ms

  if (Serial.available() > 0) {
    //Only run when it receive data

    //Convert string msg to int
    int state = Serial.readString().toInt();

    //Turn ON/OFF the lamp
    if(state == 1){
      digitalWrite(lampPin,HIGH);
    }
    else if(state == 0){
      digitalWrite(lampPin,LOW);
    }
  }
}
```

Pada tugas ini, saya mengasumsikan menggunakan photosensor Light Dependent Resistor (LDR) yang mempunyai rangkaian kurang lebih seperti berikut.



Perhitungan konversi ini dapat berubah sesuai dengan tegangan, mikrokontroller yang digunakan, serta besar resistansi yang digunakan. Tegangan dari rangkaian akan dibaca oleh mikrokontroller kemudian dikonversi menjadi satuan lux.

```
float LDR_value = analogRead(A0); //Read value from ldr
float lux = (250.00/(ADC_value*LDR_value))-50.00; //Convert from analog value into lux
```

Jika ingin menggunakan sensor temperature dan humidity lainnya dan pembacaan via analogRead, saya membuat kode yang dikomen dimana nilai akan disimpan dan kemudian di scale agar nilai tersebut menjadi dalam satuan celcius dan persentase. Satuan scaler ini dapat diubah nilai perkalian atau penjumlahan maupun pengurangan sesuai dengan spesifikasi sensor pada datasheet.

```
/*If using own temperature and humidity sensor
//Input
// int humid = analogRead(A1);
// int temp = analogRead(A2);

//Scaling for celcius and percentage
// temp_scaled = temp*scaler_temp
// humid_scaled = humid*scaler_humid/100
*/
```

Dan jika menggunakan sensor DHT11, maka dapat menggunakan library dht dimana bacaan sudah dalam satuan celcius dan persentase

```
//If using dht11 sensor
float humid = dht.readHumidity(); //Read humidity (already in percentage)
float temp = dht.readTemperature(); //Read temperature (already in celcius)
```

Bagian terakhir dari kode merupakan mengubah nilai bacaan ketiga sensor menjadi string, dan ketiga string tersebut di encode menjadi satu buah string **msg**. String **msg** tersebut kemudian dikirimkan ke kode python via serial.

```
dtostrf(lux, 7, 2, outstr_photosensor); //Convert data from float/int to string for message transmission using sprintf()
dtostrf(humid, 7, 2, outstr_humid);
dtostrf(temp, 7, 2, outstr_temp);

sprintf(msg, "%s %s %s", outstr_photosensor, outstr_humid, outstr_temp); //Encode message
Serial.println(msg); //Send message
```