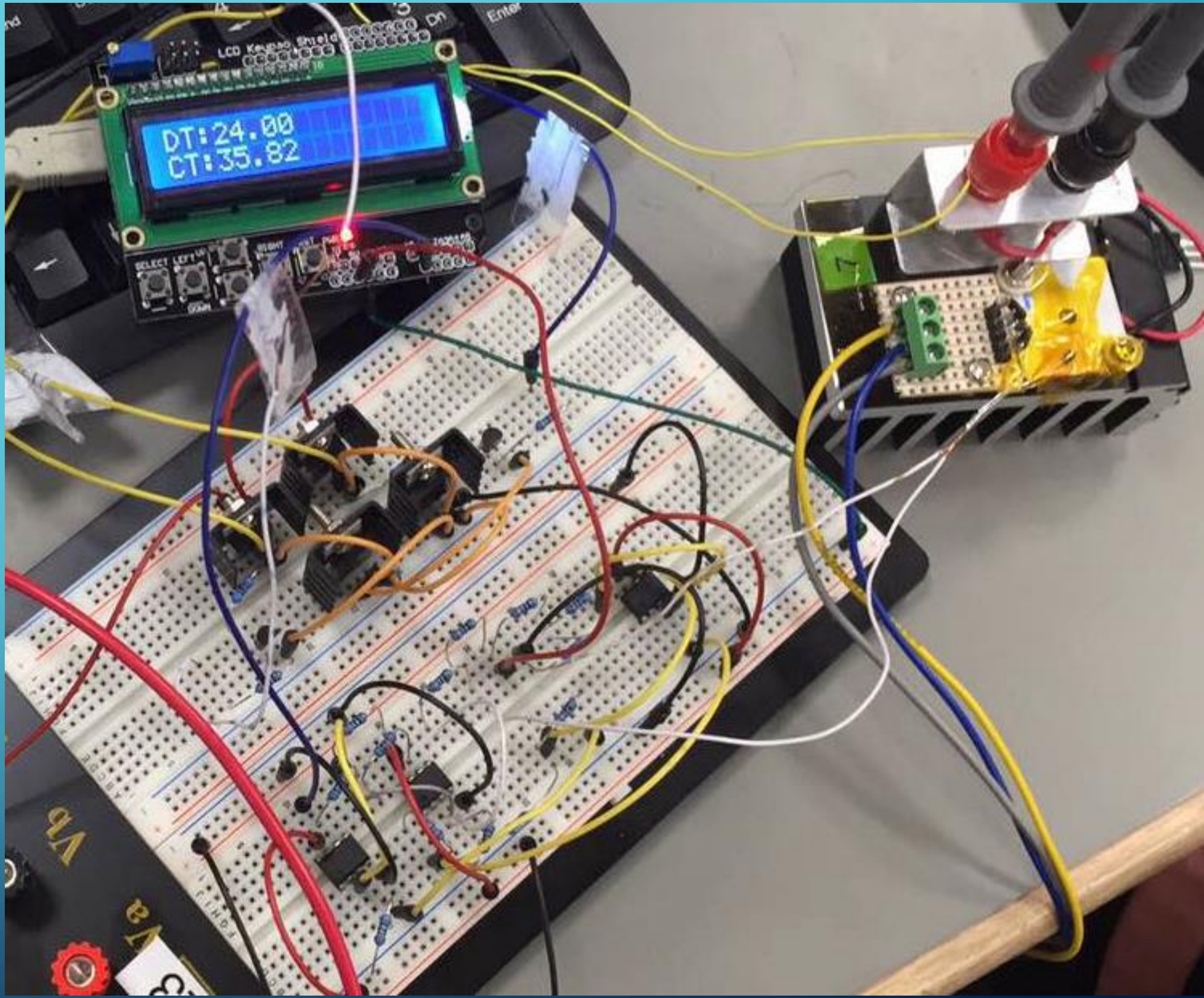
ELEN30013
ESI PRESENTATION



WEDNESDAY 3:15-6:15

GROUP 7 YUE CHANG 872301

YI JIAN 850509

OVERALL DESIGN

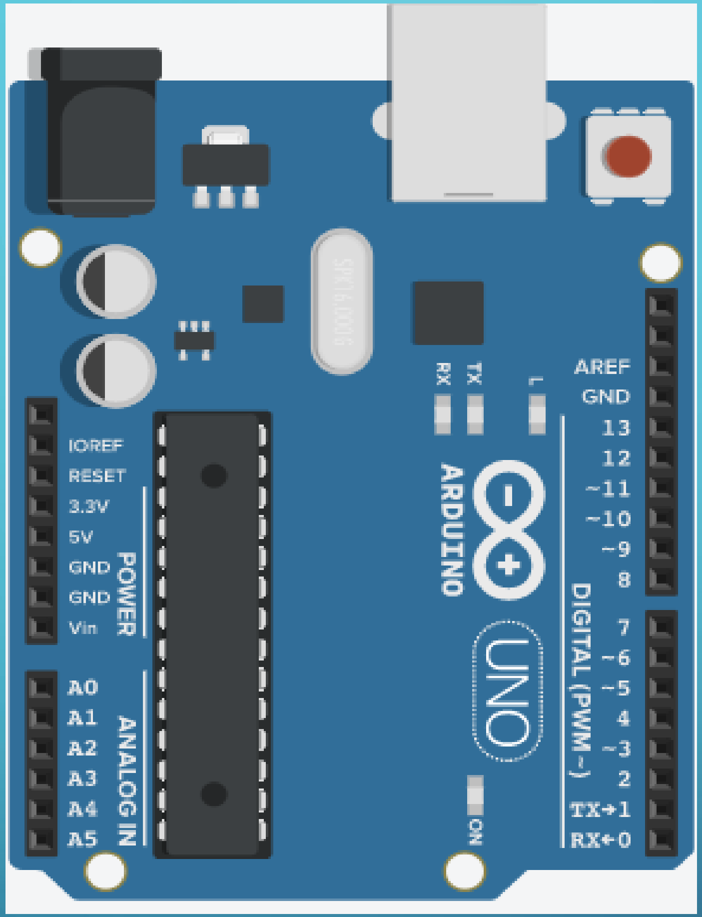
Input desired temp. DT (A0)

anologRead buttons' sense value to detect which button is pressed

Sensor signal-conditioning circuit(5V A1)

Wheatstone bridge and instrumental amplifier

anologRead A1 using the relation between current temp.CT and sensevalue A1 to compute CT



TEC Drive Circuit(D3 D11)

H-bridge+PWM

H-bridge control Itec direction

PWM control Itec size

LCD display temp. CT&DT (D4 D5 D6 D7 D8 D9 5V)

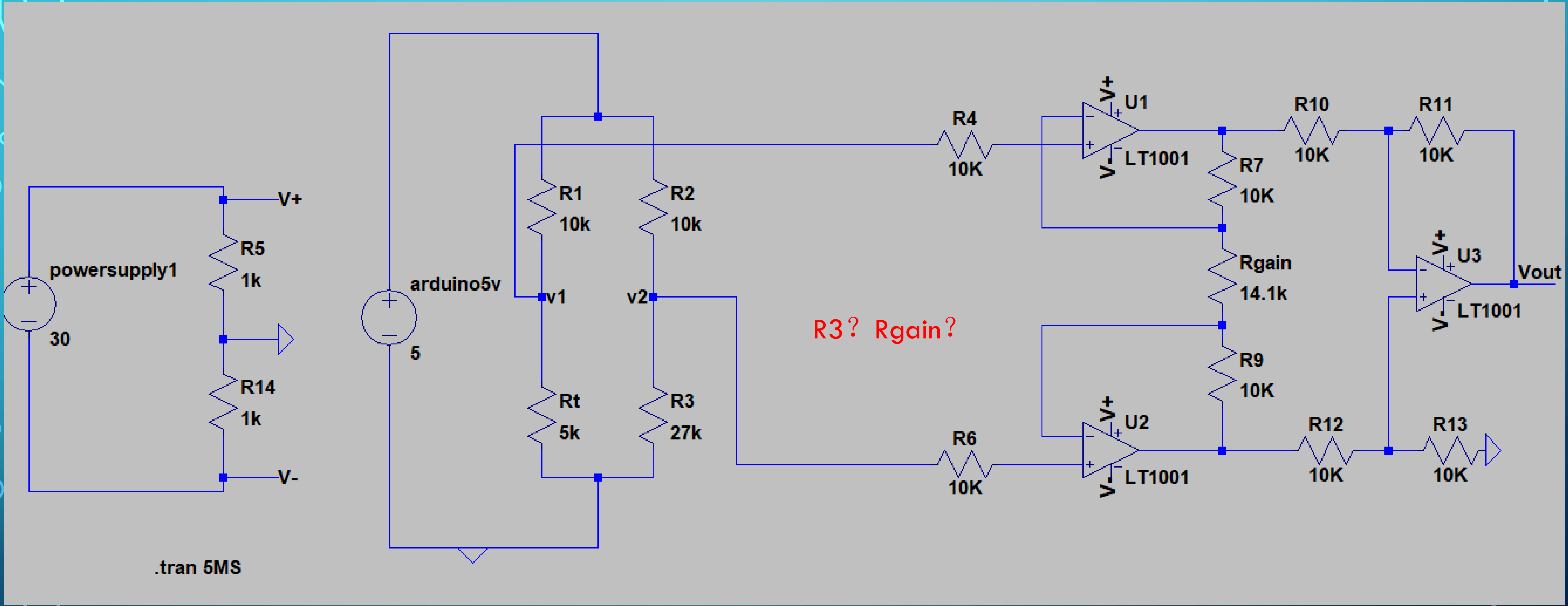
TIMELINE AND WORK DISTRIBUTION

week	Free time	Workshop time
6	Determine project requirements review lecture contents(together)	Search for information(together)
7	Design the system overall framework List time table(together)	Learning principles of sensor and driving circuit discuss and choose method(togerther)
8	Work distribution(together)	Learn and start to purchase LCD and buttons for display and input temp.(YI) Start simulation of H-bridge and Sensor circuit.(YUE)
9	Discuss and error check and modification of the simulation of h-bridge and sensor(together)	Build the physical circuit for h-bridge(YI) Error check and lab records.(YUE)
Non-teaching period	1.Running simulation for LCD and buttons(YI) 2.Arduino code for display temp. and input temp. parts(YI)	1.Modify simulation of h-bridge and sensor circuit(YUE) 2.Arduino code for sensor reading calibration and h-bridge control(YUE)

TIMELINE AND WORK DISTRIBUTION

week	Free time	Workshop time
10	Code and simulation errors check Experiments contents arrangement (together)	Open lab(Mon): Another physical circuit for h-bridge(YI),error check and lab records(YUE) Workshop: Physical circuits for h-bridge and sensor circuits(YI),error check and lab records(YUE) Open lab(Wen):Test Arduino code for h-bridge, change PWM(YUE) Record Itec and PWM for h-bridge(YI) Open lab(Fri): Modifications for Sensor circuits and test Vout(<5v) for different RT(YUE) Test LCD display and buttons for input temp.(YI)
11	Organize the code and prepare for project check(together)	Workshop: put all parts together and error check,calibration(together) Open lab(Fri):Physical test, calibration, improvements plans(together) Open lab(Mon):Test improvements plans, come to conclusion(together)

SENSOR SIGNAL CONDITIONING



- Wheatstone bridge and instrumentation amplifier

SENSOR SIGNAL CONDITIONING

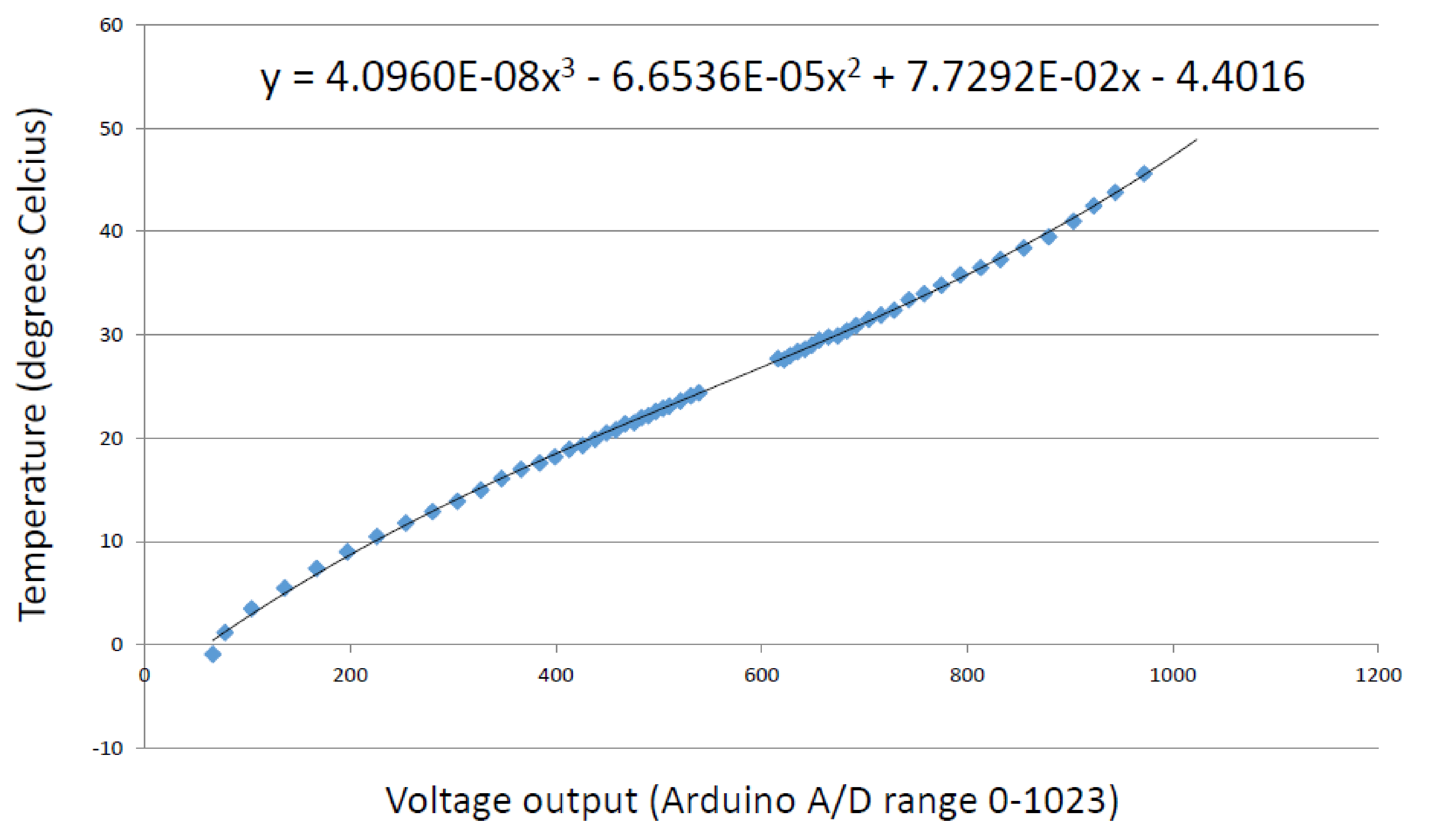
How to choose R3 and Rgain?

- **R3**: R3>RTmax（R(45°)=4.4kΩ，R(5°)=25.5kΩ） v2>v1,Vout>0 Arduino can read.
- **Rgain**: voltage difference<5v after amplified for Arduino to read. (Gain=1+20k/Rgain)

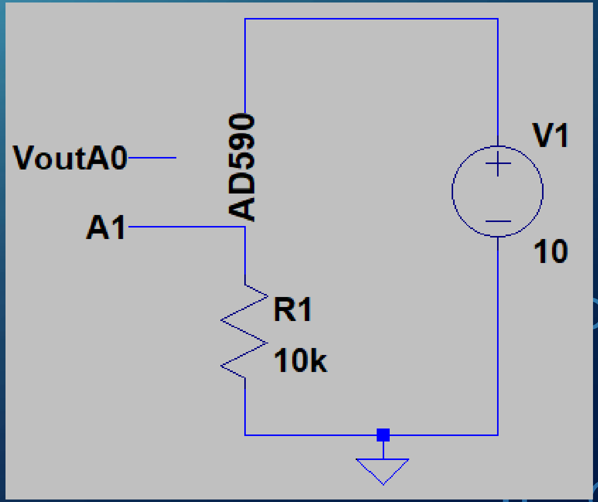
Characteristics:

- Vout 0-5v fully utilized 0-1023 of the Arduino A/D converter increase accuracy of reading.
- Noise from power supply is correlated in V1 and V2 and cancels.
- Measure temp？ CT→RT→Vout→analogRead A1→SenseValue
- Calibration： find the relationship between SenseValue and CT

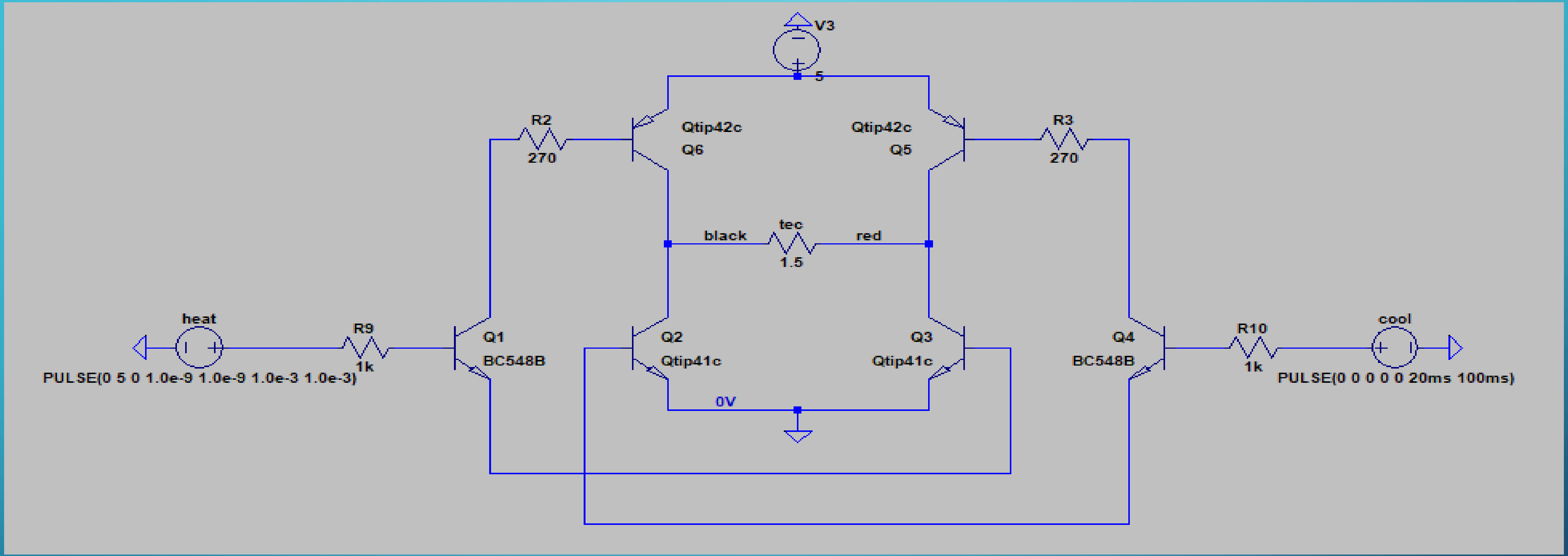
CALIBRATION




```
void loop(){
sensorValue=analogRead(A0);
Serial.print("x= ");
Serial.println(sensorValue);
sensorValue1=analogRead(A1);
Serial.println(sensorValue1);
CT=1000*(((sensorValue1/1023*5)/10)-0.2732);
Serial.print("y=");
Serial.println(CT,4);
}
```



TEC DRIVING CIRCUIT



Status	ON	OFF
Heat	Q6(Saturation)Q1(Saturation)Q3(Linear)	Q5 Q2 Q4
Cool	Q5(Saturation)Q2(Saturation)Q4(Linear)	Q6 Q1 Q3

- H-bridge using PWM to control.

TEC DRIVER CIRCUIT

Need to consider two things:

① **DT-25**(assume room temp=25° to compute absolute value)

- DT>>room temp, need larger current to reach DT
- DT around room temp, use small current to avoid over cool/heat
- Through measurements, function between DT-25 and pwm output
- F(x) = -0.243*x*x+10.31*x+11.260504201680671

② **DT-CT**

- Large temp difference need large current to speed up
- Small temp difference need small current to keep steady

PWM

control

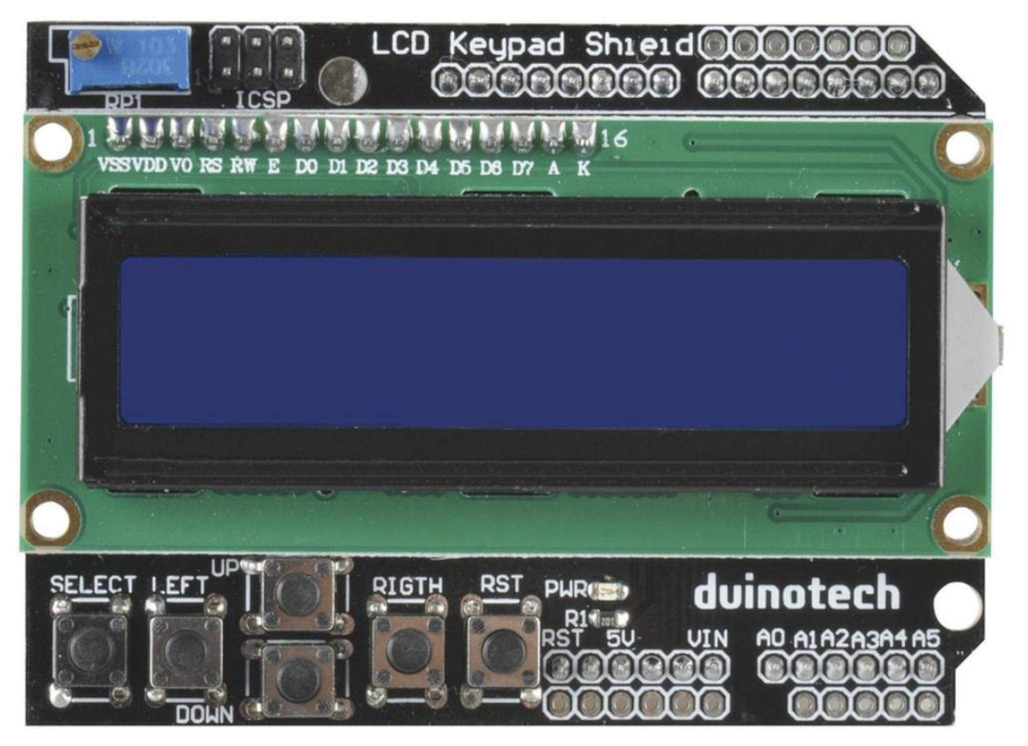
ARDUINO CODE (SENSOR AND H-BRIDGE)

```
//Read CT(current temp.)

sensorValue = analogRead(A1);

CT = (4.096E-08)*pow(sensorValue, 3) - (6.6536E-05)*pow(sensorValue, 2) + 0.077292*sensorValue - 4.4016;
//Relationship between sensorvalue and temp.

//code for controlling h-bridge

if ((-0.5)<(CT - DT) && (CT - DT)<0.5) {

digitalWrite(3, LOW);

digitalWrite(11, LOW);

Serial.print("don't need cool or heat");}

if (CT>0.5 + DT)//need cooling{

A = (int)(CT - DT + 0.5);  //The temperature difference is rounded
Serial.println(A);}
      switch (A) {
      case 1://For small DT-CT
C = (int)(fabs(DT - 25)); //Absolute value
D = (int)((-0.244)*pow(C, 2) + 10.311*C + 11.261); //Relationship between DT-25 and pwm output
analogWrite(3, D);//output for cool pin
      …
      case 35://For large DT-CT
analogWrite(3, 220); //big constant pwm output for cool pin
delay(500);
break;
```

LCD CONTROLLER MODULE

Specifications	
Operating Voltage	5VDC
PushsButton Protocol	Analog
LCD Size	16 Columns x 2 Rows
Additional Features	Backlight

Pinout	
D9	E
D*	RS
D7	DB7
D6	DB6
D5	DB5
D4	DB4
A0	Buttons



To initalise the shield for use with the Liquid Crystal Library use the Liquid Crystal LCD(8,9,4,5,6,7);

ARDUINO CODE (LCD—SETUP)

```
#include <LiquidCrystal.h> //LCD1602的库
double CT = 12.34567;
double DT;
int lastkey = 0;     //which key was pressed last cycle
unsigned long t0;    //how long has key been held for
          //pin for buttons
#define KEYPIN A0
          //button constants
#define btnRIGHT  6
#define btnUP     5
#define btnDOWN   4
#define btnLEFT   3
#define btnSELECT 2
#define btnNONE   (-1)
LiquidCrystal lcd1602(8, 9, 4, 5, 6, 7);//Set the Arduino interface with the LCD screen
void setup()
{
  Serial.begin(9600);
  lcd1602.begin(16, 2);//Initialize the LCD
}
```

Include the library of LCD1602.
Setup variables for test.

Pin setup.

Initialize the LCD

ARDUINO CODE (LCD—READ SUBROUTINE)

```
int read_LCD_buttons()
{
  int adc_key_in = 0;
  adc_key_in = analogRead(KEYPIN);      // read the value from the sensor
  delay(5);
//switch debounce delay. Increase this delay if incorrect switch selections are returned.
  int k = (analogRead(KEYPIN) - adc_key_in);
//gives the button a slight range to allow for a little contact resistance noise
  if (5 < abs(k)) return btnNONE;
// double checks the keypress. If the two readings are not equal +/-k value after debounce delay, it tries again.

  if (adc_key_in > 1000) return btnNONE; // We make this the 1st option for speed reasons since it will be the most likely result
  if (adc_key_in <50)  return btnRIGHT;//
  if (adc_key_in <195)  return btnUP;
  if (adc_key_in <380)  return btnDOWN;
  if (adc_key_in <555)  return btnLEFT;
  if (adc_key_in <790)  return btnSELECT;
  return btnRIGHT;
}
```

When a button is pushed down, it will produce an analog voltage to Arduino Pin A0.

We need to double check to make sure that no buttons have been accidentally touched.

Read the analog value from Arduino to determine which button is pushed down.

ARDUINO CODE (LCD—DO SUBROUTINE)

```
void dobuttons()      //updates variables. debounces by only sampling at intervals
{
  int key;
  key = read_LCD_buttons();
  switch (key) {
  case btnLEFT:
    DT = DT - 10;
    delay(500);
    break;
  case btnRIGHT:
    DT = DT + 10;
    delay(500);
    break;
  case btnUP:
    DT = DT + 0.5;
    delay(500);
    break;
  case btnDOWN:
    DT = DT - 0.5;
    delay(500);
    break;
  case btnSELECT:
    DT = DT+10;
    delay(500);
    break;
  case btnNONE:
    DT = DT + 10;
    delay(500);
    break;
  }
}
```

After we determine which button is pressed, the corresponding instructions need to be executed.

In our project,
when the LEFT button is pushed, the desired temperature will minus 10 ;
when the RIGHT button is pushed, the desired temperature will plus 10;
when the UP button is pushed, the desired temperature will plus 0.5;
when the DOWN button is pushed, the desired temperature will miuns 0.5;

ARDUINO CODE (LCD—MAIN)

```
void loop()
{
  int key = read_LCD_buttons();         //read keys
  lastkey = key;                        //save for next
  unsigned int t;
  if ((key>0) && (lastkey == 0))      //if key pressed
    {
    t0 = millis();                //record when button pressed
    dobuttons();                  //act once straight away
    }
if ((key>0) && (lastkey == key))   //if button is still held pressed
    {
    if (millis() - t0>400)
      {
      t0 = t0 + 200;      //repeat after delay if held down
      dobuttons();
       }
     };

//Display content on LCD screen
  lcd1602.clear();
  lcd1602.setCursor(0, 0);//Move the cursor to the first row
  lcd1602.print("DT:");//The content shows on the first row
  lcd1602.print(DT);//Show the number of DT
  lcd1602.setCursor(0, 1);//Move the cursor to the second row
  lcd1602.print("CT:");//The content shows on the second row
  lcd1602.print(CT, 2);//Show the number of CT
}
```

1.Calling subroutine-read_LCD_buttons(), determine which key is pressed down.

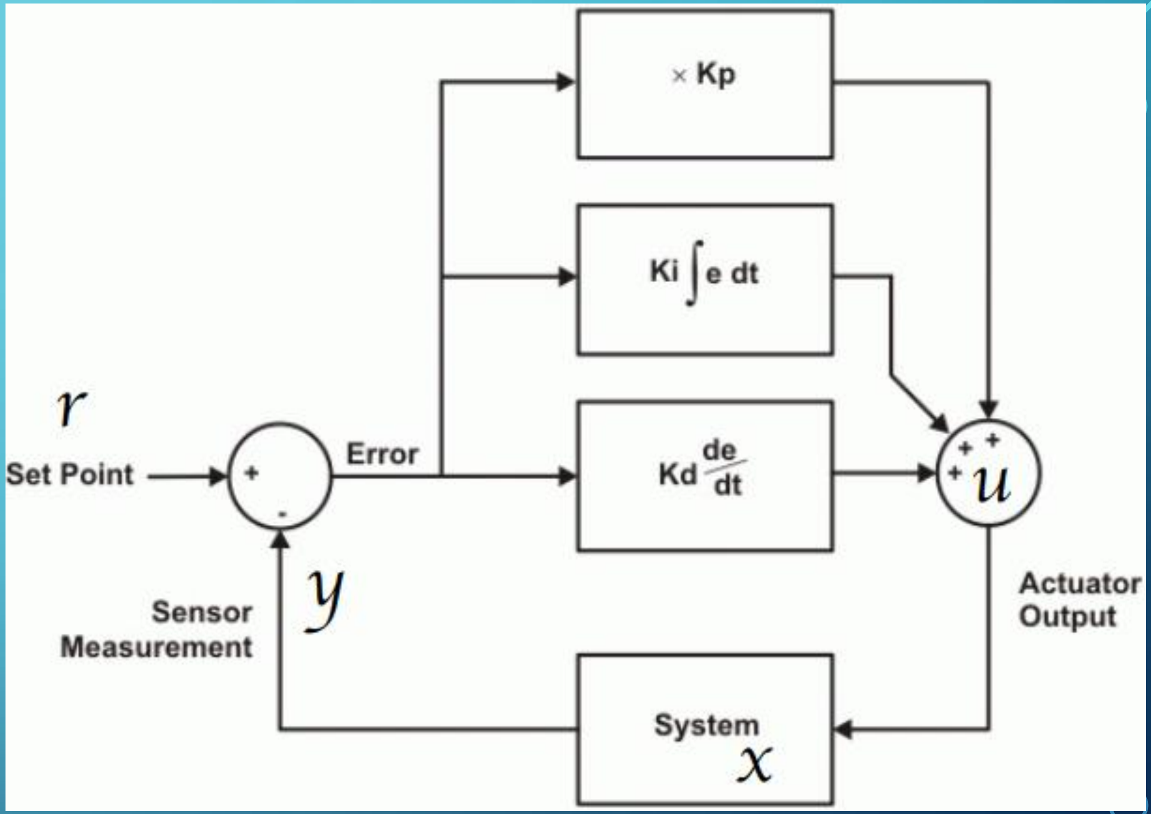
2.Whether the button is pressed or being held pressed. Calling subroutine-dobuttons()-to make DT change. If the button is still held pressed, repeat the change.

3. Display content on LCD screen.
First row:DT:**.**
Second row:CT:**.**

SYSTEM IMPROVEMENT Ⅰ

- When the set temperature and room temperature difference is large, we need to provide high current to maintain the temperature of TEC, but also it is likely to cause greater error.

  →Use PID controller to precisely control the temperature

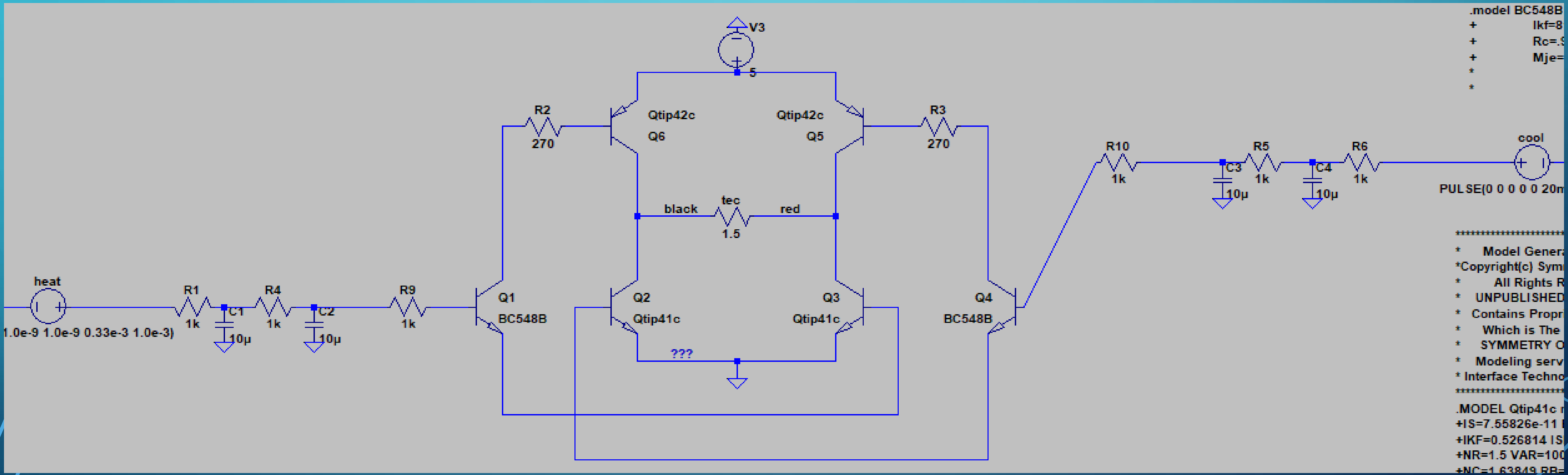


- Schematic diagram of PID control

SYSTEM IMPROVEMENT II

- We use PWM to control the output voltage from Arduino

  →Add low pass filter circuits to get a fixed analog value



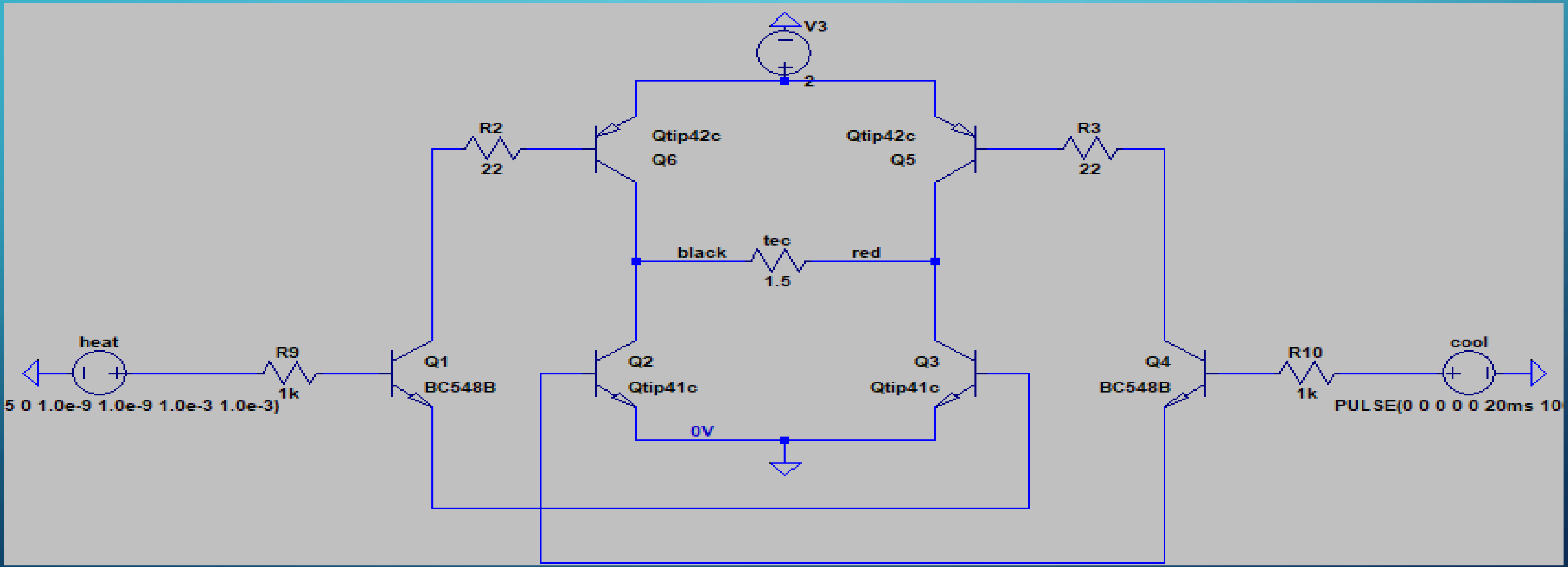
- H-bridge with 2nd order low pass filter

SYSTEM IMPROVEMENT III

- The transistors in H-bridge can easily get hot and burned out

→Change the voltage or resistance to make all transistors to work in saturation region.

→Soldered all components onto a piece of Vero board



- Another design of H-bridge that all transistors work in saturation region

CONCLUSION

System performance:

- Range: 5° -45°
- Accuracy: +/-0.5
- Settling time: <2mins

Driving circuit and sensor circuit features:

- I tec range: 0-932mA
- Vout sensevalue range: 60-990 (approx.)

Power consumption:

- Sensor circuit: 0.83w
- Driving circuit: 4.7w