The Hall Effect In Semiconductors

Burak Kara
Boğaziçi University • Physics Department

***Abstract*—In the presence of a magnetic field, a current-carrying conductor's charge carriers feel a force due to the Lorentz Force, the force pushes carriers to the sides of the material, leading to a voltage difference. This voltage difference is called Hall Voltage. It was first realized by Edwin Hall in 1879. The Hall Voltage depends on the applied magnetic field and the current, but also it depends on the intrinsic properties of the material itself. The intrinsic property dependence is called the Hall Coefficient, and it is beneficial to determine the characteristics of the materials. In this experiment, an n-doped Germanium is tested with varying magnetic fields and currents. The Hall Coefficient is found to be: $R_H = -0.00714 \pm 0.00085 m^3/A.s$, and the mobility of the charge carriers (electrons) is found to be: $\mu = -0.259 \pm 0.058 m^2/V.s$.**

I. Introduction

The Hall Effect in conductors produces a voltage difference between two sides of the conductor in the presence of a magnetic field with a current flow on the conductor. The voltage is called "Hall Voltage", and it is proportional to the applied magnetic field, applied current, and properties of the material. It was first realized by Edwin Hall in 1879[1]. This proportionality may be also defined by a coefficient, which is the Hall Coefficient. The coefficient is proportional to the number of charge carriers, and the type of carriers. The type dependence of the coefficient yields different signs for semiconductors depending on the structure of the material [2]. Therefore, it is beneficial for determining the characteristics of the material.

The semiconductor is a material in which its conductivity is in between conductors and insulators. The main property of semiconductors different than other materials is their electron configuration. More precisely, the gaps in their electronic excitation spectra. The most known semiconductors are the elements Silicon(Si) and Germanium(Ge) which are elemental semiconductors and have 4 valance electrons [3]. Moreover, their conductivity can be increased with "doping". The doping is the process of introducing a different compound to the lattice structure to create an intentional impurity. This impurity leads to extra electrons or fewer electrons depending on the introduced material. If the added material has 5 valance electrons, then it produces a negative(n) type of doping. If the material has 3 valance electrons, then it produces a positive(p) type of doping [4]. The charge carriers are "electrons" and "holes" respectively[5]. Depending on the dope type, it is expected to see different signs of Hall Coefficients. In this experiment n-doped Germanium is used, therefore it is expected to see a negative Hall Coefficient, which is validated.

II. Theory

If a magnetic field is applied to a current-carrying conductor, the magnetic field leads to the Lorentz Force on the charge carriers. The holes and electrons will build up on the two opposite sides, therefore leading to a voltage difference. This voltage difference is called Hall Voltage. Also, the induced electric field yields a Lorentz Force too. In the steady state, these forces are equal, therefore the Hall Coefficient and the Hall Voltage can be obtained by the Drude Model.[6]

Induced Electric Field:
$$[2] E_H = R_H \cdot j_\perp \cdot B \tag{1}$$

The Hall Coefficient:
$$[2] R_H = \frac{1}{ne} \tag{2}$$

Hall Coefficient for Mixed Charge Carriers:
$$[2] R_{\mathrm{H}} = \frac{n_h \mu_{\mathrm{h}}^2 - n_e \mu_{\mathrm{e}}^2}{e(n_h \mu_{\mathrm{h}} + n_e \mu_{\mathrm{e}})^2} \tag{3}$$

The Hall Voltage:
$$[6] V_H = \frac{IB}{tne} \tag{4}$$

Combining Eq. 2 and 4:
$$R_H = \frac{V_H t}{IB} \tag{5}$$

Conductivity:
$$[6] \sigma = \frac{L}{AR} = e(n_e \mu_e + n_h \mu_h) \tag{6}$$

Mobility (Assuming Majority is Electrons):
$$\mu_e = \frac{L}{AR e n_e} = \frac{R_H L}{AR} \tag{7}$$

where j is the current density, n is the carrier concentration,
μ is the mobility of carriers, t is the thickness,
A is the cross-section, L is the Length, R is the resistance,
I is the current, B is the magnetic field and e is the elementary charge.
$e = 1.602176634 \times 10^{-19} C [7]$

For the derivations of the equations above, see The Oxford Solid State Basics[6].

III. The Experimental Setup & Method

For the experiment the following apparatus are used:

- Semiconductor Hall Probe with holder and stand,
- n-doped Germanium,
- Helmholtz Coils,
- Teslameter,
- Digital Multimeter,
- Power Supply,
- Connecting Leads,
- Magnetic Compass,
- Ruler.

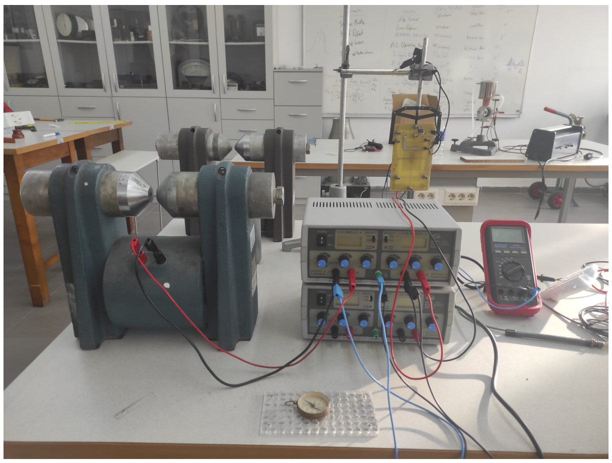

Fig. 1. Apparatus.

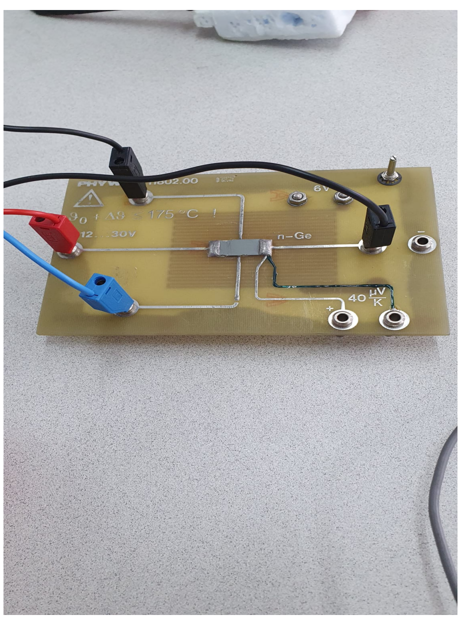

Fig. 2. n-dope Germanium Hall Probe.

For the experiment the following procedure is followed:

1) The dimensions of the material are measured
   Length: $16 \pm 4mm$ (Current Direction),
   Width: $10 \pm 1mm$ (Hall Voltage Direction),
   Thickness: $1.0 \pm 0.2mm$,
2) The resistance of the Hall Probe is measured,
3) The Hall Voltage is measured without the magnetic field for reference (normally it is not expected to see a Hall Voltage when the magnetic field is not present; however, due to the asymmetry on the probe, there exists a voltage difference.),
4) Teslameter is adjusted to obtain 0 magnetic field reading,
5) Hall Voltage is recorded with a constant current. Both negative and positive magnetic field is used,
6) Hall Voltage is recorded with a constant magnetic field, with varying current.

IV. The Data

The measurements for the experiment are listed in this section.

TABLE I
Resistance Measurements

Current(mA)	Voltage(V)
10. ± 1.	0.431 ± 0.024
15. ± 1.	0.659 ± 0.036
20. ± 1.	0.872 ± 0.047
25. ± 1.	1.111 ± 0.058
30. ± 1.	1.311 ± 0.068
35. ± 1.	1.538 ± 0.080
40. ± 1.	1.754 ± 0.091
45. ± 1.	1.978 ± 0.102
50. ± 1.	2.202 ± 0.113
55. ± 1.	2.417 ± 0.124

TABLE II
Reference Measurements

Current(mA)	Hall Voltage(mV)
5. ± 1.	3.8 ± 0.2
10. ± 1.	6.9 ± 0.4
15. ± 1.	10.1 ± 0.5
20. ± 1.	13.5 ± 0.7
25. ± 1.	17.2 ± 0.9
30. ± 1.	20.4 ± 1.1
35. ± 1.	24.3 ± 1.2
40. ± 1.	27.2 ± 1.4
45. ± 1.	31.1 ± 1.6
50. ± 1.	34.4 ± 1.8
55. ± 1.	38.1 ± 1.9
60. ± 1.	41.7 ± 2.4

TABLE III
Positive Magnetic Field Measurements
Semi-Conductor Current = $35 \pm 1mA$

Coil Current (A)	Magnetic Field (mT)	Hall Voltage (mV)
0.05 ± 0.05	18.2 ± 0.1	19.4 ± 1.0
0.10 ± 0.05	31.6 ± 0.1	15.9 ± 0.8
0.15 ± 0.05	46.3 ± 0.1	12.2 ± 0.6
0.20 ± 0.05	61.8 ± 0.1	8.4 ± 0.5
0.25 ± 0.05	76.3 ± 0.1	4.6 ± 0.3
0.30 ± 0.05	90.0 ± 0.1	1.1 ± 0.1
0.35 ± 0.05	104.5 ± 0.1	-2.6 ± 0.1
0.40 ± 0.05	120.5 ± 0.1	-6.5 ± 0.4
0.45 ± 0.05	135.4 ± 0.1	-10.2 ± 0.5
0.50 ± 0.05	150.6 ± 0.1	-14.1 ± 0.7
0.55 ± 0.05	165.3 ± 0.1	-17.7 ± 0.9
0.60 ± 0.05	180.8 ± 0.1	-21.6 ± 1.1

TABLE IV
NEGATIVE MAGNETIC FIELD MEASUREMENTS
SEMI-CONDUCTOR CURRENT = 35 $\pm$ $1mA$

Coil Current (A)	Magnetic Field (mT)	Hall Voltage (mV)
-0.05 $\pm$ 0.05	-12.3 $\pm$ 0.1	27.1 $\pm$ 1.4
-0.10 $\pm$ 0.05	-25.2 $\pm$ 0.1	30.4 $\pm$ 1.6
-0.15 $\pm$ 0.05	-40.9 $\pm$ 0.1	34.4 $\pm$ 1.8
-0.20 $\pm$ 0.05	-56.4 $\pm$ 0.1	38.3 $\pm$ 1.9
-0.25 $\pm$ 0.05	-71.2 $\pm$ 0.1	42.1 $\pm$ 2.4
-0.30 $\pm$ 0.05	-89.7 $\pm$ 0.1	46.8 $\pm$ 2.6
-0.35 $\pm$ 0.05	-103.8 $\pm$ 0.1	50.4 $\pm$ 2.8
-0.40 $\pm$ 0.05	-117.9 $\pm$ 0.1	53.8 $\pm$ 3.0
-0.45 $\pm$ 0.05	-132.2 $\pm$ 0.1	57.5 $\pm$ 3.2
-0.50 $\pm$ 0.05	-146.6 $\pm$ 0.1	61.2 $\pm$ 3.4
-0.55 $\pm$ 0.05	-163.5 $\pm$ 0.1	65.4 $\pm$ 3.6
-0.60 $\pm$ 0.05	-178.7 $\pm$ 0.1	69.2 $\pm$ 3.8

TABLE V
VARYING SEMI-CONDUCTOR CURRENT MEASUREMENTS
COIL CURRENT = -0.3 $\pm$0.1 A
MAGNETIC FIELD = -88.5 $\pm$ 0,1 mT

Semi-Conductor Current (mA)	Hall Voltage (mV)
5. $\pm$1.	7.4 $\pm$0.4
10. $\pm$1.	13.1 $\pm$0.7
15. $\pm$1.	20.2 $\pm$1.0
20. $\pm$1.	26.9 $\pm$1.4
25. $\pm$1.	33.2 $\pm$1.7
30. $\pm$1.	39.2 $\pm$2.0
35. $\pm$1.	46.7 $\pm$2.6
40. $\pm$1.	52.8 $\pm$2.9
45. $\pm$1.	59.2 $\pm$3.3
50. $\pm$1.	66.0 $\pm$3.6
55. $\pm$1.	72.9 $\pm$3.9
60. $\pm$1.	80.2 $\pm$4.3

V. THE ANALYSIS AND RESULTS

The following equation is considered for the error propagation, assuming there is no correlation between variables.

$$\sigma_f = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 \sigma_x^2 + \left(\frac{\partial f}{\partial y}\right)^2 \sigma_y^2 + \left(\frac{\partial f}{\partial z}\right)^2 \sigma_z^2 + \cdots} \quad (11)$$

The error propagation of the digital multimeter is done according to the manual of the manufacturer. The results extracted from the graphs' errors are directly obtained from the ROOT TF1 Class. The uncertainties of the readings other than the digital multimeter are taken as the least significant figures. For the implementation of the error propagation formula, see the appendix.

From TableI, the voltage vs current plot is fitted by ROOT, and the resistance is obtained from the slope of the line.

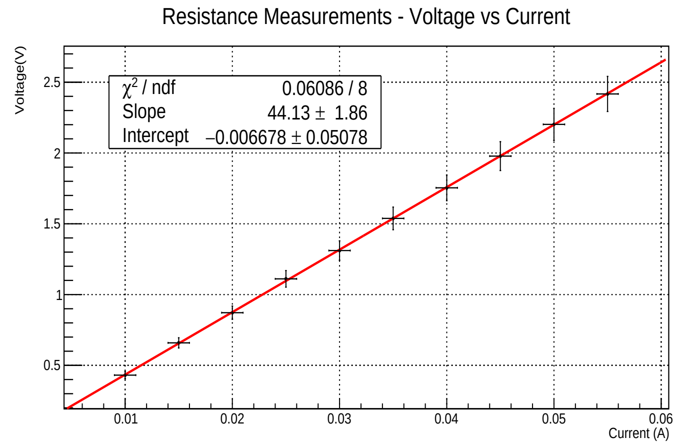

Fig. 3. Resistance Measurement.

The resistance is calculated as $44.13 \pm 1.86\Omega$.

As mentioned in the Method section, the reference measurements without a magnetic field are made and listed in Table.II. The Voltage and current are plotted and fitted to a line to obtain an offset equation for varying current measurements.

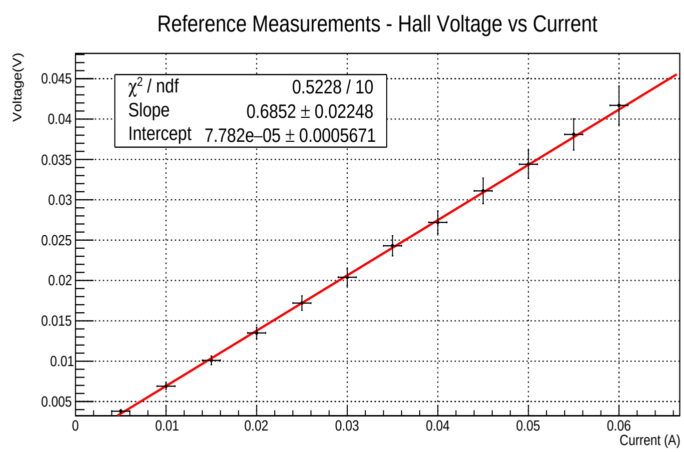

Fig. 4. Reference Measurements.

From Fig.4, the obtained offset equation is $V = 0.6852I \pm 0.2248$. The intercept is omitted since the error is higher than the values itself.

From Table.III the voltage and magnetic field data is fitted to a line. The offset is not needed for this measurement, since the current does not vary and the slope is needed for the calculation.

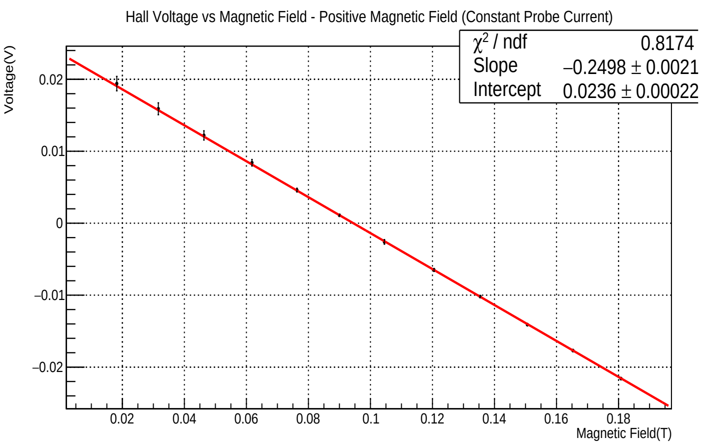

Fig. 5. Positive Magnetic Field Measurements.

As can be seen from the figure, the Slope(V/T) is $-0.250 \pm 0.002V/T$. From equation 5, the Hall Coefficient can be calculated as $\dfrac{Slope \cdot t}{I}$. which yields to: $R_H = -0.00714 \pm 0.00144m^3/A.s$. According to equation 7 and the dimensions of the material given in the Experimental Setup section, mobility can be calculated using resistance and Hall coefficient. The resulting mobility is $\mu = -0.259 \pm 0.099m^2/V.s$.

From Table.IV the voltage and magnetic field data is fitted to a line. Similar to positive magnetic field measurements, the offset is not needed.

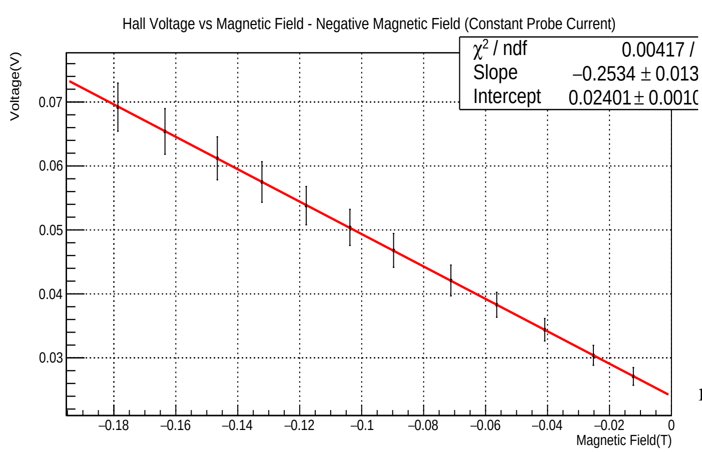

Fig. 6. Negative Magnetic Field Measurements.

Similar to the positive magnetic field measurements, the Hall Coefficient is calculated as $R_H = -0.00724 \pm 0.00151m^3/A.s$. The mobility is calculated as $\mu = -0.262 \pm 0.101m^2/V.s$.

From Table.V, the voltage and current data are fitted to a line. Since the current varies, the offset is needed in this measurement. From reference measurements, the obtained offset equation is $V = 0.6852I \pm 0.2248$. Subtracting the measured voltage from Table.V, and fitting a line gives the following figures:

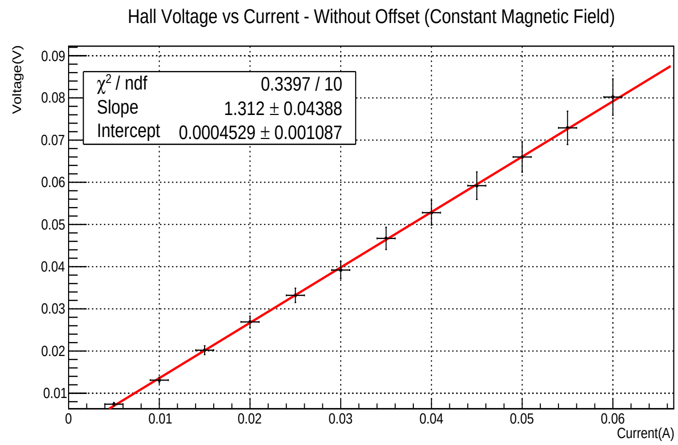

Fig. 7. Varying Current Measurements Without Offset.

As can be seen from the figure, the Slope(V/I) is $1.312 \pm 0.044V/A$. From equation 5, the Hall Coefficient can be calculated as $\dfrac{Slope \cdot t}{B}$. which yields to: $R_H = -0.0148 \pm 0.0030m^3/A.s$. According to equation 7, the mobility is calculated as $\mu = -0.538 \pm 0.205m^2/V.s$. Comparing the values to previous calculations shows the importance of the offset.

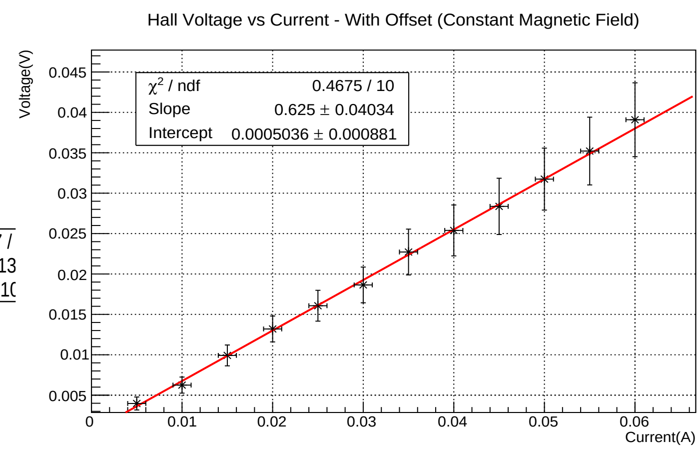

Fig. 8. Varying Current Measurements - With Offset.

Similar to the calculations of the without offset measurement, the Hall Coefficient is calculated as:

$R_H = -0.00706 \pm 0.00148cm^3/A.s$. The mobility is calculated as $\mu = -0.256 \pm 0.099m^2/V.s$. These values are in agreement with the previous values.

VI. The Conclusion

The Hall Coefficient for a n-doped Germanium is calculated with varying current and varying magnetic field. Three different measurements are made which are:

- Constant Current - Varying Positive Magnetic Field
  $R_H = -0.00714 \pm 0.00144m^3/A.s$.
  $\mu = -0.259 \pm 0.099m^2/V.s$.

- Constant Current - Varying Negative Magnetic Field
  $R_H = -0.00724 \pm 0.00151m^3/A.s$.

$\mu = -0.262 \pm 0.101 m^2/V.s.$

- Constant Negative Magnetic Field - Varying Current
  $R_H = R_H = -0.00706 \pm 0.00148 cm^3/A.s.$
  $\mu = -0.256 \pm 0.099 m^2/V.s.$

Lastly, the weighted average of the obtained values is taken to extract a single measurement. Which gives: $R_H = -0.00714 \pm 0.00085 m^3/A.s$, and $\mu = -0.259 \pm 0.058 m^2/V.s$. If these values are considered as the true values, then the previous calculations are 0σ, 0.066σ, and 0.054σ away respectively from the true value for R_H calculation. Similarly, for mobility calculations, they are 0σ, 0.029σ, and 0.030σ away respectively from the true value.

All values are in less than one sigma away from each other. Therefore, the experiment was a success. However, the uncertainties are considerably large, this is due to the precision of the apparatus that is used. A better resolution should lead to a better result. Moreover, the effects need further experiments with a p-doped semiconductor and a conductor.

References

[1] *Doped Semiconductors*. URL: http://hyperphysics.phy-astr.gsu.edu/hbase/Solids/dope.html (visited on 12/16/2023).

[2] E. Gülmez. *Advanced Physics Experiments*. 1st. Boğaziçi University Publications, 1999.

[3] Manuel Cardona Peter Y. Yu. *Fundamentals of Semiconductors: Physics and Materials Properties*. 4th. Springer Berlin, Heidelberg, 2010. DOI: https://doi.org/10.1007/978-3-642-00710-1.

[4] *On a New Action of the Magnet on Electric Currents*. URL: https://web.archive.org/web/20110727010116/http://www.stenomuseet.dk/skoletj/elmag/kilde9.html (visited on 12/16/2023).

[5] *Charge Carrier*. URL: https://energyeducation.ca/encyclopedia/Semiconductor (visited on 12/16/2023).

[6] Steven H Simon. *The Oxford solid state basics*. Oxford, UK: Oxford Univ. Press, 2013. URL: https://cds.cern.ch/record/1581455.

[7] *CODATA Value: elementary charge*. URL: https://physics.nist.gov/cgi-bin/cuu/Value?e (visited on 11/24/2023).

VII. Appendix

We have used ROOT Version: 6.28/04 on Ubuntu 22.04.2 LTS machine for the following macros.

functions.h Header File:

```
#ifndef functions
#define functions

#include <math.h>
#include <vector>
#include <string>
#include <iostream>
#include <iomanip>
using namespace std;

void printResult(double val, double error, int
    sf)
{
   int factor = sf - ceil(log10(fabs(error)));
   if (factor <= 0)
   {
      cout << setprecision(sf) << val;
      cout << " +- " << error << endl;
   }
   else
   {
      cout << fixed << setprecision(factor) <<
          val;
      cout << " +- " << error << endl;
   }
}

double calculateVoltUncertainty(double V)
{
   V = abs(V);
   double error;
   if (V <= 0.4)
   {
      error = (3 * V / 100) + 0.4;
   }
   else if (V <= 4)
   {
      error = (5 * V / 100) + 0.003;
   }
   else if (V <= 40)
   {
      error = (5 * V / 100) + 0.03;
   }
   else if (V <= 400)
   {
      error = (5 * V / 100) + 0.3;
   }
   else
   {
      error = (V / 100) + 4;
   }
   return error;
};


double calculateAmpUncertainty(double A)
{
   A = abs(A);
   return (2 * A / 100) + 0.005;
};


double calculateLogError(double val, double
    sigma)
{
   return sigma / val;
};
#endif
```

Resistance and Reference Calculations:

```
#include "functions.h"
using namespace std;

void part0()
{
   // Initialize:
   gStyle->SetOptFit(1);
   auto *linefit = new TF1("linefit", "[0]*x +
       [1]");
```

```
auto *graph1 = new TGraphErrors();
auto *graph2 = new TGraphErrors();

linefit->SetParNames("Slope", "Intercept");
graph1->SetTitle("Resistance Measurements -
    Voltage vs Current;Current (A);Voltage(V
    )");
graph2->SetTitle("Reference Measurements -
    Hall Voltage vs Current;Current (A);
    Voltage(V)");

vector<double> Probe_V_resistance =
    {0.431,0.659,0.872,1.111,
1.311,1.538,1.754,1.978,2.202,2.417};// Volt
vector<double> Probe_I_resistance =
    {10.000,15.000,20.000,25.000,
30.000,35.000,40.000,45.000,50.000,55.000};
    // mA
vector<double> Probe_V_bias =
    {3.80,6.90,10.10,13.50,17.20,
20.40,24.30,27.20,31.10,34.40,38.10,41.70};
    //mV
vector<double> Probe_I_bias =
    {5,10,15,20,25,
30,35,40,45,50,55,60};//mA

double I_error = 1e-3;
double convert = 1e-3;

// Fill Graph1:
for (size_t i = 0; i < Probe_V_resistance.
    size(); i++)
{
   double V = Probe_V_resistance[i];
   double I = Probe_I_resistance[i] *
       convert;

   double V_error = calculateVoltUncertainty
       (V);

   graph1->SetPoint(i, I, V);
   graph1->SetPointError(i, I_error, V_error
       );
}

// Fill Graph2:
for (size_t i = 0; i < Probe_V_bias.size();
    i++)
{
   double V = Probe_V_bias[i] * convert;
   double I = Probe_I_bias[i] * convert;

   double V_error = calculateVoltUncertainty
       (Probe_V_bias[i])*convert;

   graph2->SetPoint(i, I, V);
   graph2->SetPointError(i, I_error, V_error
       );
}

// Fit Graph1:
linefit->SetRange(0,1);
auto *c1 = new TCanvas("c1", "c1", 200, 10,
    600, 400);
c1->SetGrid();
c1->Draw();
graph1->Fit("linefit");
graph1->Draw ("A*");

   // Print Result:
   double R = linefit->GetParameter(0);
   double R_error = linefit->GetParError(0);
   cout << "R = ";
   printResult(R, R_error, 3);

   // Fit Grah2:
   linefit->SetRange(0,1);
   auto *c2 = new TCanvas("c2", "c2", 200, 10,
       600, 400);
   c2->SetGrid();
   c2->Draw();
   graph2->Fit("linefit");
   graph2->Draw("A*");

   // Print Result:
   double slope = linefit->GetParameter(0);
   double intercept = linefit->GetParameter(1);
   double slope_error = linefit->GetParError(0)
       ;
   double intercept_error = linefit->
       GetParError(1);
   cout << "Slope = ";
   printResult(slope, slope_error, 3);
   cout << "Intercept = ";
   printResult(intercept, intercept_error, 3);
};
```

Hall Coefficient, and mobility calculations:

```
#include "functions.h"
using namespace std;

void part1()
{
   // Initialize:
   gStyle->SetOptFit(1);
   auto *linefit = new TF1("linefit", "[0]*x +
       [1]");
   auto *graph1 = new TGraphErrors();
   auto *graph2 = new TGraphErrors();
   auto *graph3 = new TGraphErrors();

   linefit->SetParNames("Slope", "Intercept");
   graph1->SetTitle("Hall Voltage vs Magnetic
       Field - Positive Magnetic Field (
       Constant Probe Current); Magnetic Field(
       T); Voltage(V)");
   graph2->SetTitle("Hall Voltage vs Magnetic
       Field - Negative Magnetic Field (
       Constant Probe Current); Magnetic Field(
       T); Voltage(V)");
   graph3->SetTitle("Hall Voltage vs Current -
       With Offset (Constant Magnetic Field);
       Current(A); Voltage(V)");

   vector<double> measure1_B =
       {18.2000,31.6000,46.3000,61.8000,
   76.3000,90.0000,104.5000,120.5000,
   135.4000,150.6000,165.3000,180.8000,};
   vector<double> measure1_V =
       {19.4000,15.9000,12.2000,8.4000,
   4.6000,1.1000,-2.6000,-6.5000,
   -10.2000,-14.1000,-17.7000,-21.6000};
   double measure1_I = 0.035;

   vector<double> measure2_B =
       {-12.3000,-25.2000,-40.9000,-56.4000,
```

```
-71.2000,-89.7000,-103.8000,-117.9000,
-132.2000,-146.6000,-163.5000,-178.7000,};
vector<double> measure2_V =
    {27.1000,30.4000,34.4000,38.3000,
42.1000,46.8000,50.4000,53.8000,
57.5000,61.2000,65.4000,69.2000};
double measure2_I = 0.035;

vector<double> measure3_I =
    {5.0000,10.0000,15.0000,20.0000,
25.0000,30.0000,35.0000,40.0000,
45.0000,50.0000,55.0000,60.0000};
vector<double> measure3_V =
    {7.4000,13.1000,20.2000,26.9000,
33.2000,39.2000,46.7000,52.8000,
59.2000,66.0000,72.9000,80.2000};
double measure3_B = -88.5e-3;

double B_error = 5/100;
double I_error = 1e-3;
double convert = 1e-3;
double t = 1e-3;
double t_error = 0.2e-3;
double L = 16e-3;
double L_error = 4e-3;
double w = 10e-3;
double w_error = 0.2e-3;
double A = w*t;
double A_error = sqrt(pow(w_error*t,2)+pow(w
    *t_error,2));
double R = 44.13;
double R_error = 1.86;

double sigma = L/(A*R);
double sigma_error = sqrt(pow(L_error/(A*R)
    ,2)+pow(A_error*sigma/A,2)+pow(R_error*
    sigma/R,2));

double bias_slope = 0.6852;
double bias_slope_error = 0.0225;

// Fill Graph1:
for (size_t i = 0; i < measure1_V.size(); i
    ++)
{
   double V = measure1_V[i] * convert;
   double B = measure1_B[i] * convert;

   double V_error = calculateVoltUncertainty
       (measure1_V[i])*convert;
   double B_error_true = B*B_error;

   graph1->SetPoint(i, B, V);
   graph1->SetPointError(i, B_error_true,
       V_error);
}

// Fill Graph2:
for (size_t i = 0; i < measure2_V.size(); i
    ++)
{
   double V = measure2_V[i] * convert;
   double B = measure2_B[i] * convert;

   double V_error = calculateVoltUncertainty
       (measure2_V[i])*convert;
   double B_error_true = B*B_error;

   graph2->SetPoint(i, B, V);
   graph2->SetPointError(i, B_error_true,
       V_error);
}

// Fill Graph3:
for (size_t i = 0; i < measure3_V.size(); i
    ++)
{
   double V = measure3_V[i] * convert;
   double I = measure3_I[i] * convert;

   double bias = bias_slope*I;
   V -= bias;

   double V_error_temp =
       calculateVoltUncertainty(measure3_V[i
       ])*convert;
   double V_error = sqrt(pow(V_error_temp,2)
       +pow(I_error*bias_slope,2)+pow(
       bias_slope_error*I,2));

   graph3->SetPoint(i, I, V);
   graph3->SetPointError(i, I_error, V_error
       );
}

// Fit Graph1:
auto *c1 = new TCanvas("c1", "c1", 200, 10,
    600, 400);
c1->SetGrid();
c1->Draw();
graph1->Fit("linefit","Q");
graph1->Draw ("A*");

// Print Result:
double slope = linefit->GetParameter(0);
double intercept = linefit->GetParameter(1);
double slope_error = linefit->GetParError(0)
    ;
double intercept_error = linefit->
    GetParError(1);
cout << "Positive Magnetic Field:" << endl;
cout << "Slope = ";
printResult(slope, slope_error, 3);
cout << "Intercept = ";
printResult(intercept, intercept_error, 3);

// Calculate Hall Coefficient
double R_H = slope*t/measure1_I;
double R_H_error = sqrt(pow(slope_error*t/
    measure1_I,2) + pow(slope*t_error/
    measure1_I,2) + pow(I_error*slope*t/(pow
    (measure1_I,2)),2));
cout << "R_H = ";
printResult(R_H,R_H_error,3);

double mobility = R_H*sigma;
double mobility_error = sqrt(pow(R_H_error*
    sigma,2)+pow(R_H * sigma_error,2));
cout << "Mobility = ";
printResult(mobility,mobility_error,3);

// Fit Graph2:
auto *c2 = new TCanvas("c2", "c2", 200, 10,
    600, 400);
c2->SetGrid();
c2->Draw();
graph2->Fit("linefit","Q");
graph2->Draw ("A*");
```

```
    // Print Result:
    slope = linefit->GetParameter(0);
    intercept = linefit->GetParameter(1);
    slope_error = linefit->GetParError(0);
    intercept_error = linefit->GetParError(1);
    cout << "Negative Magnetic Field:" << endl;
    cout << "Slope = ";
    printResult(slope, slope_error, 3);
    cout << "Intercept = ";
    printResult(intercept, intercept_error, 3);

    // Calculate Hall Coefficient
    R_H = slope*t/measure2_I;
    R_H_error = sqrt(pow(slope_error*t/
        measure2_I,2) + pow(slope*t_error/
        measure2_I,2) + pow(I_error*slope*t/(pow
        (measure2_I,2)),2));
    cout << "R_H = ";
    printResult(R_H,R_H_error,3);

    //Calculate Mobility
    mobility = R_H*sigma;
    mobility_error = sqrt(pow(R_H_error*sigma,2)
        +pow(R_H * sigma_error,2));
    cout << "Mobility = ";
    printResult(mobility,mobility_error,3);

    // Fit Graph3:
    auto *c3 = new TCanvas("c3", "c3", 200, 10,
        600, 400);
    c3->SetGrid();
    c3->Draw();
    graph3->Fit("linefit","Q");
    graph3->Draw ("A*");

    // Print Result:
    slope = linefit->GetParameter(0);
    intercept = linefit->GetParameter(1);
    slope_error = linefit->GetParError(0);
    intercept_error = linefit->GetParError(1);
    cout << "Varying Current:" << endl;
    cout << "Slope = ";
    printResult(slope, slope_error, 3);
    cout << "Intercept = ";
    printResult(intercept, intercept_error, 3);

    // Calculate Hall Coefficient
    R_H = slope*t/measure3_B;
    double B_error_true = measure3_B*B_error;
    R_H_error = sqrt(pow(slope_error*t/
        measure3_B,2) + pow(slope*t_error/
        measure3_B,2) + pow(B_error_true*slope*t
        /(pow(measure3_B,2)),2));
    cout << "R_H = ";
    printResult(R_H,R_H_error,3);

    // Calculate Mobility
    mobility = R_H*sigma;
    mobility_error = sqrt(pow(R_H_error*sigma,2)
        +pow(R_H * sigma_error,2));
    cout << "Mobility = ";
    printResult(mobility,mobility_error,3);
};
```