**ĐẠI HỌC QUỐC GIA TP. HCM**

**TRƯỜNG ĐẠI HỌC BÁCH KHOA**



**BÀI TẬP MỞ RỘNG MÔN HỌC ĐO LƯỜNG CÔNG NGHIỆP**

**ĐỀ TÀI**

**XÂY DỰNG GIAO DIỆN MÁY TÍNH, THIẾT KẾ HỆ THỐNG THU THẬP NHIỆT ĐỘ TỪ 2 CẢM BIẾN NHIỆT LM35 VÀ 1 THERMISTOR**

**Lớp: L02 - HK222**

**GVHD: NGUYỄN ĐỨC HOÀNG**

**SINH VIÊN THỰC HIỆN**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **MSSV** | **HỌ** | **TÊN** | **ĐIỂM** | **GHI CHÚ** |
| 1 | 2010327 | Lê Mỹ | Khánh |  |  |
| 2 | 2010298 | Võ Ngọc Nhất | Huy |  |  |
| 3 | 2010469 | Võ Dương Xuân | Nguyên |  |  |

**­­­­­­­­­­**

**MỤC LỤC**

[**1. Phần cứng** 2](#_Toc136772032)

[**2. Phần mềm** 3](#_Toc136772033)

[**3. Giao diện** 13](#_Toc136772034)

[**4. Kết quả chạy:** 13](#_Toc136772035)

# **1. Phần cứng**

Các mạch gồm các phần tử như sau:

- 1 pin nguồn 9V, 1 mạch ổn nguồn 5V

- 1 điện trở 5.1k, 2 điện trở 4.7k, 2 điện trở 1k.

- 4 đèn LED.

- 2 cảm biến nhiệt độ LM35

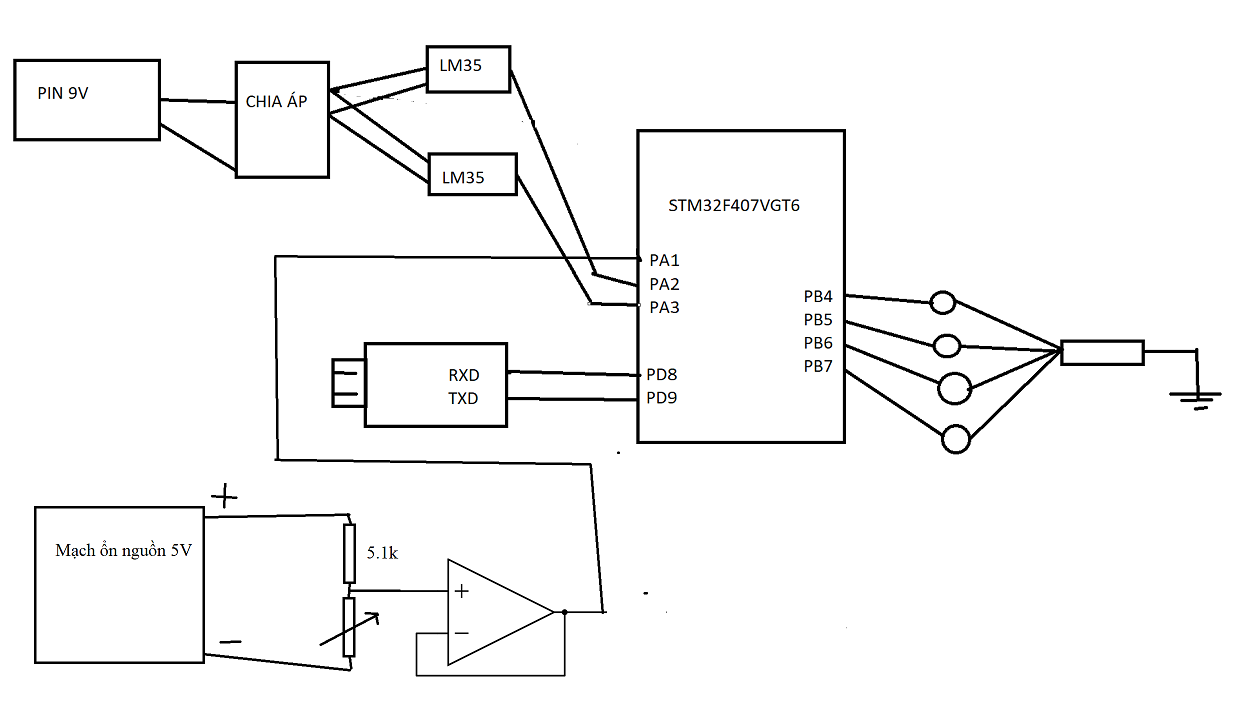
- 1 thermistor NTC có thông số beta = 3950, R25 = 5kΩ

- 1 board STM32F407VGT6

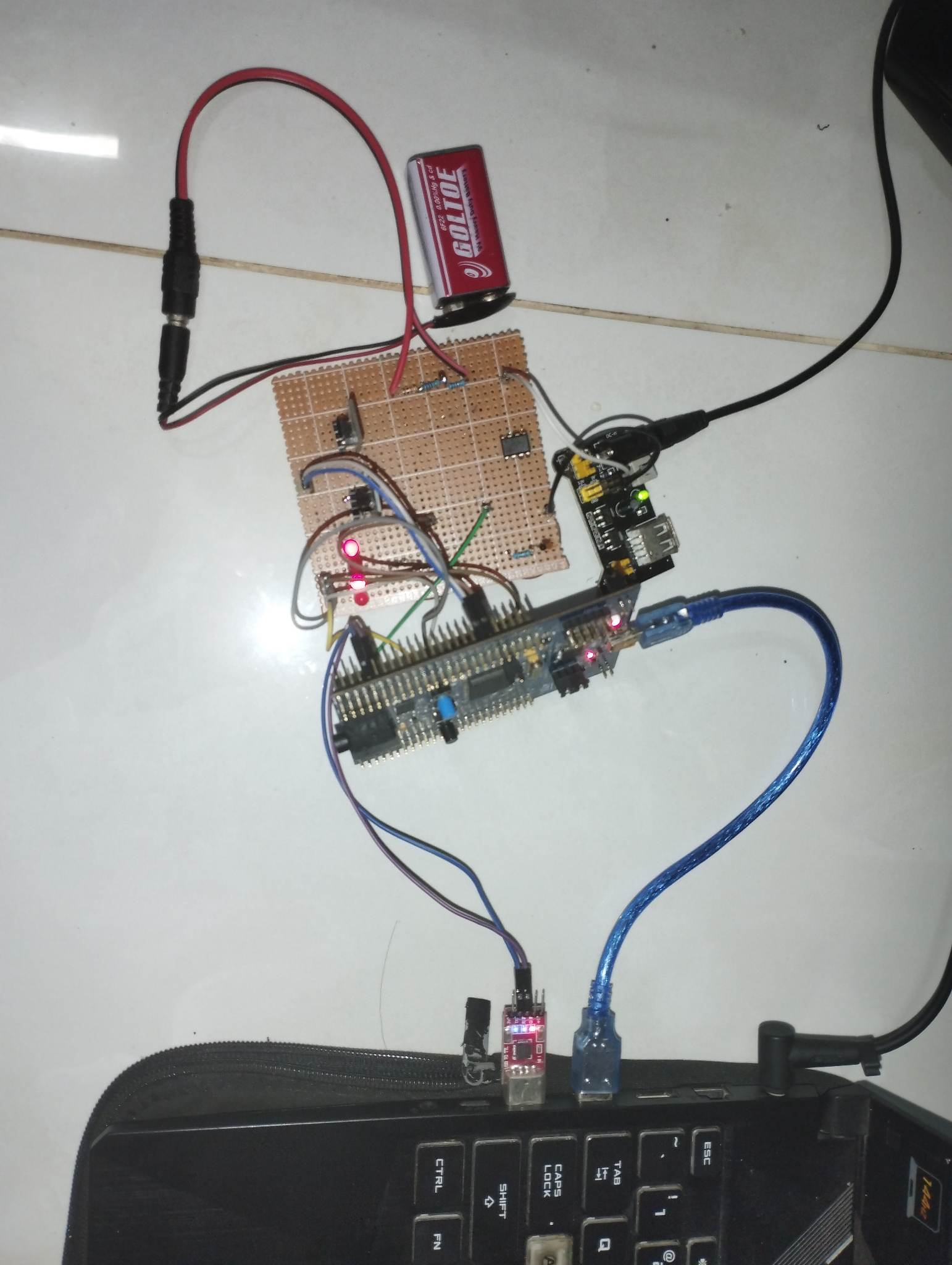
- 1 USB TTL.

- Dây bus, miếng đồng đục lỗ sẵn.

Mạch nguyên lí:



Mạch thực tế:



# **2. Phần mềm**

Code thực hiện các phần sau:

- Cấp xung Clock cho GPIO, các ngoại vi ADC và USART, DMA.

- Cấu hình GPIO (4 chân hiển thị LED Threshold, 3 chân đọc ADC và 2 chân Tx, Rx để truyền nhận dữ liệu từ máy tính thông qua USB TTL), cấu hình DMA, ADC (12 bit), UART (truyền dữ liệu bằng DMA).

- Sử dụng RTOS để phân chia chương trình thành 3 Task:

+ Task 1: Đọc và xử lý giá trị ADC, đưa dữ liệu vào buffer TX\_Buffer để chuẩn bị truyền lên laptop (Frame truyền: 20 bytes, 4 bytes đầu điện áp đọc từ thermistor, 4 bytes điện áp đọc từ LM35 số 1 và 4 bytes điện áp đọc từ LM35 số 2, 8 bytes cuối là “\r\n”).

+ Task 2: Gửi giá trị ADC đã xử lý trong TX\_Buffer từ STM32 lên laptop thông qua USART3 sử dụng DMA1 Stream3.

+ Task 3: Truyền dữ liệu tín hiệu đèn từ GUI về vi xử lý vào buffer RX\_Buffer của DMA để hiển thị tín hiệu đèn thông qua DMA1 Stream1. (Frame truyền: 8 bytes “AxByCzDt”, với x, y, z và t có thể là “0” – tắt đèn, “1” – bật đèn, các kí tự “A”, “B”, “C”, “D” tượng trưng cho mỗi LED).

Lưu ý: Giữa Task 2 và Task 3 sử dụng Binary Semaphore để ngăn không cho 2 Task đọc – ghi đồng thời làm nhiễu loạn thông tin của thanh ghi Data Register của USART3.

Code STM32F407VGT6:

- Cấu hình ADC hoạt động ở chế độ Independent với độ phân giải 12 bit như sau:

|  |
| --- |
| hadc3.Instance = ADC3;  hadc3.Init.ClockPrescaler = ADC\_CLOCK\_SYNC\_PCLK\_DIV2;  hadc3.Init.Resolution = ADC\_RESOLUTION\_12B;  hadc3.Init.ScanConvMode = DISABLE;  hadc3.Init.ContinuousConvMode = DISABLE;  hadc3.Init.DiscontinuousConvMode = DISABLE;  hadc3.Init.ExternalTrigConvEdge = ADC\_EXTERNALTRIGCONVEDGE\_NONE;  hadc3.Init.ExternalTrigConv = ADC\_SOFTWARE\_START;  hadc3.Init.DataAlign = ADC\_DATAALIGN\_RIGHT;  hadc3.Init.NbrOfConversion = 1;  hadc3.Init.DMAContinuousRequests = DISABLE;  hadc3.Init.EOCSelection = ADC\_EOC\_SINGLE\_CONV; |

- Cấu hình USART3 hoạt động ở tốc độ Baud 9600:

|  |
| --- |
| huart3.Instance = USART3;  huart3.Init.BaudRate = 9600;  huart3.Init.WordLength = UART\_WORDLENGTH\_8B;  huart3.Init.StopBits = UART\_STOPBITS\_1;  huart3.Init.Parity = UART\_PARITY\_NONE;  huart3.Init.Mode = UART\_MODE\_TX\_RX;  huart3.Init.HwFlowCtl = UART\_HWCONTROL\_NONE;  huart3.Init.OverSampling = UART\_OVERSAMPLING\_16; |

- Cấu hình GPIOB4🡪GPIOB7 ở mode output để xuất tín hiệu đèn LED báo ngưỡng:

|  |
| --- |
| GPIO\_InitStruct.Pin = GPIO\_PIN\_4|GPIO\_PIN\_5|GPIO\_PIN\_6|GPIO\_PIN\_7;  GPIO\_InitStruct.Mode = GPIO\_MODE\_OUTPUT\_PP;  GPIO\_InitStruct.Pull = GPIO\_NOPULL;  GPIO\_InitStruct.Speed = GPIO\_SPEED\_FREQ\_LOW;  HAL\_GPIO\_Init(GPIOB, &GPIO\_InitStruct); |

- Task 1: Đọc ADC và đưa vào Buffer

|  |
| --- |
| void Task1\_ADC(void const \* argument)  {  /\* USER CODE BEGIN 5 \*/  /\* Infinite loop \*/  for(;;)  {  HAL\_ADC\_Start(&hadc1);  HAL\_ADC\_PollForConversion(&hadc1, 100);  TX\_Buffer[0] = Process\_value(HAL\_ADC\_GetValue(&hadc1) \* 5000/0xFFF);  HAL\_ADC\_Stop(&hadc1);  HAL\_ADC\_Start(&hadc2);  HAL\_ADC\_PollForConversion(&hadc2, 100);  TX\_Buffer[1] = Process\_value(HAL\_ADC\_GetValue(&hadc2) \* 5000/0xFFF);  HAL\_ADC\_Stop(&hadc2);  HAL\_ADC\_Start(&hadc3);  HAL\_ADC\_PollForConversion(&hadc3, 100);  TX\_Buffer[2] = Process\_value(HAL\_ADC\_GetValue(&hadc3) \* 5000/0xFFF);  HAL\_ADC\_Stop(&hadc3);  TX\_Buffer[3] = '\r';  TX\_Buffer[4] = '\n';  osDelay(100);  }  /\* USER CODE END 5 \*/  } |

- Task 2: Gửi dữ liệu ADC từ STM lên laptop

|  |
| --- |
| void Task2\_TX(void const \* argument)  {  /\* USER CODE BEGIN Task2\_TX \*/  /\* Infinite loop \*/  for(;;)  {  osSemaphoreWait(USART\_SEMAHandle, osWaitForever);  HAL\_UART\_Transmit\_DMA(&huart3, (uint8\_t \*)TX\_Buffer, sizeof(TX\_Buffer));  osSemaphoreRelease(USART\_SEMAHandle);  osDelay(1000);  }  /\* USER CODE END Task2\_TX \*/  } |

- Task 3: Đọc dữ liệu nhận từ GUI và xử lý

|  |
| --- |
| void Task3\_RX(void const \* argument)  {  /\* USER CODE BEGIN Task3\_RX \*/  /\* Infinite loop \*/  for(;;)  {  osSemaphoreWait(USART\_SEMAHandle, osWaitForever);  HAL\_UART\_Receive\_DMA(&huart3, (uint8\_t \*)RX\_Buffer, sizeof(RX\_Buffer));  if (RX\_Buffer[1] == '1') HAL\_GPIO\_WritePin(GPIOB,GPIO\_PIN\_6,GPIO\_PIN\_SET);  else HAL\_GPIO\_WritePin(GPIOB,GPIO\_PIN\_6,GPIO\_PIN\_RESET);  if (RX\_Buffer[3] == '1') HAL\_GPIO\_WritePin(GPIOB,GPIO\_PIN\_7,GPIO\_PIN\_SET);  else HAL\_GPIO\_WritePin(GPIOB,GPIO\_PIN\_7,GPIO\_PIN\_RESET);  if (RX\_Buffer[5] == '1') HAL\_GPIO\_WritePin(GPIOB,GPIO\_PIN\_4,GPIO\_PIN\_SET);  else HAL\_GPIO\_WritePin(GPIOB,GPIO\_PIN\_4,GPIO\_PIN\_RESET);  if (RX\_Buffer[7] == '1') HAL\_GPIO\_WritePin(GPIOB,GPIO\_PIN\_5,GPIO\_PIN\_SET);  else HAL\_GPIO\_WritePin(GPIOB,GPIO\_PIN\_5,GPIO\_PIN\_RESET);  osSemaphoreRelease(USART\_SEMAHandle);  osDelay(100);  }  /\* USER CODE END Task3\_RX \*/  } |

Việc tính toán nhiệt độ từ thermistor sẽ thực hiện trên code C# bằng phần mềm Visual Studio như sau:

- Tính toán giá trị điện trở từ mạch chia áp:



- Tính toán giá trị nhiệt độ từ công thức thể hiện mối liên hệ điện trở và nhiệt độ của thermistor như sau:  với .

Suy ra 

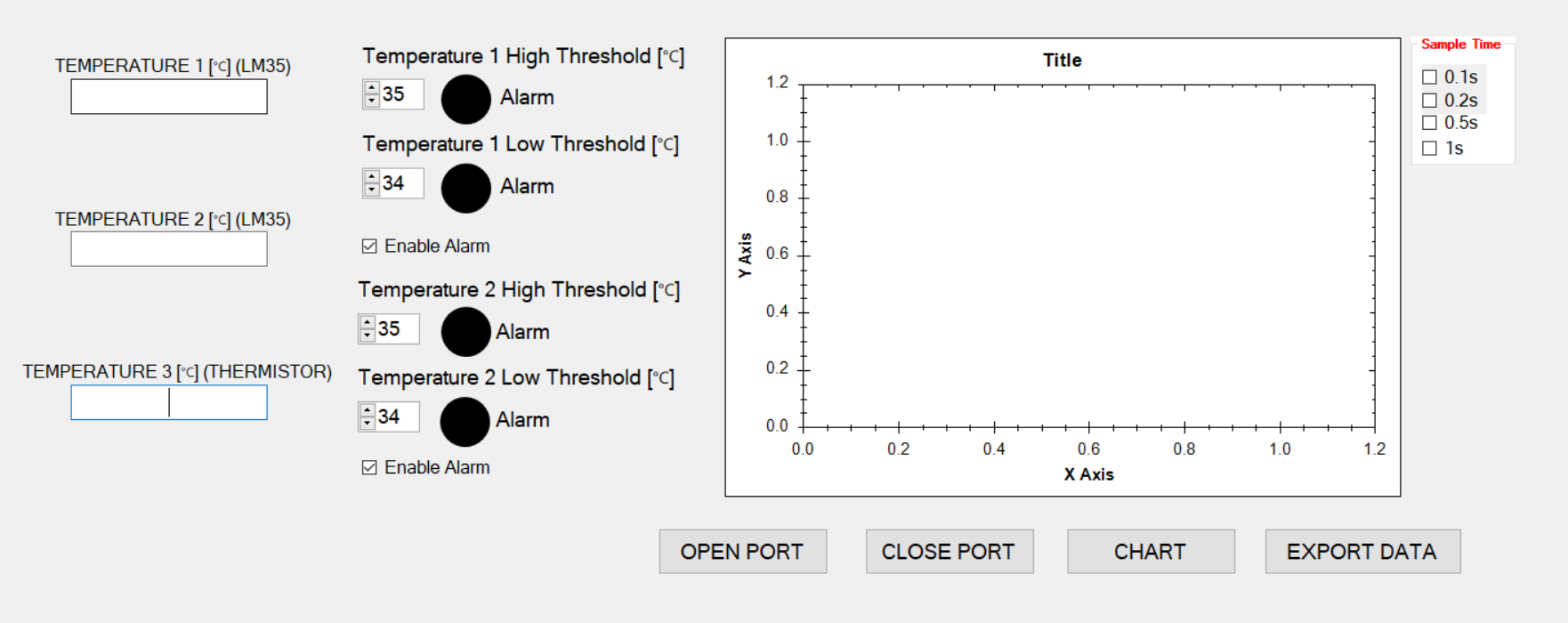
Từ đó tính được 

Đoạn code thực hiện tính toán trên C# như sau:

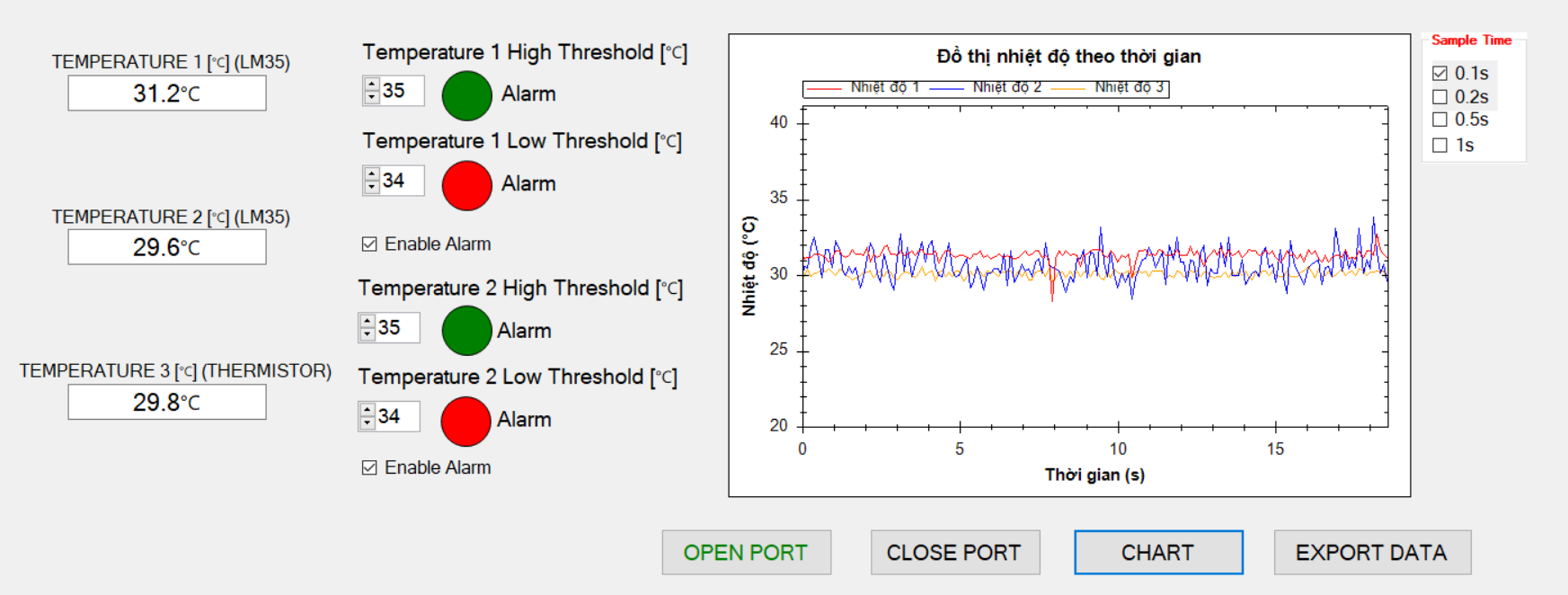
\*output\_0 là 4 byte dữ liệu đầu tiên, ứng với dữ liệu đọc được trên ADC1:

|  |
| --- |
| double ther25 = 5.0;  double beta = 3950.0;  double R1 = 5.1;  double offset = 1.4;  double ther\_vol = double.Parse(output\_0);  ther\_vol /= 1000.0;  ther\_vol -= offset;  double ther\_res = R1/(5.0/ther\_vol - 1);  double ther\_temp = System.Math.Log(ther\_res / ther25)/beta + 1/(25+273.15);  ther\_temp = 1 / ther\_temp - 273.15;  Thermistor = Math.Round(ther\_temp, 2);  Data\_Received.Text = Thermistor.ToString(); |

# **3. Giao diện**



# **4. Kết quả chạy:**



Dữ liệu được xuất ra file excel như dưới đây:

