



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



***CONTROL OPEN LOOP PADA EVAPORATOR FAN DAN ELECTRICAL  
EXPANSION VALVE (EEV) SERTA MONITOR NTC PADA AC BUS LISTRIK  
BERBASIS STM32F407***

**TUGAS AKHIR**

**Ibnu Miftah Zaini**  
**2003321060**  
**POLITEKNIK  
NEGERI  
JAKARTA**

**PROGRAM STUDI ELEKTRONIKA INDUSTRI  
JURUSAN TEKNIK ELEKTRO  
POLITEKNIK NEGERI JAKARTA  
2023**



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkannya dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



***CONTROL OPEN LOOP PADA EVAPORATOR FAN DAN ELECTRICAL EXPANSION VALVE (EEV) SERTA MONITOR NTC PADA AC BUS LISTRIK BERBASIS STM32F407***

**“Control Open Loop Stepper Motor Electric Expansion Valve Menggunakan Komunikasi Canbus Berbasis Mikrokontroler STM32F407”**

**TUGAS AKHIR**

**Diajukan sebagai salah satu syarat untuk memperoleh gelar Diploma Tiga**  
**Ibnu Miftah Zaini**  
**2003321060**

**PROGRAM STUDI ELEKTRONIKA INDUSTRI**  
**JURUSAN TEKNIK ELEKTRO**  
**POLITEKNIK NEGERI JAKARTA**

**2023**



### HALAMAN PERNYATAAN ORISINALITAS

Tugas Akhir ini adalah hasil karya saya sendiri dan semua sumber baik yang dikutip maupun dirujuk telah saya nyatakan dengan benar.

Nama : Ibnu Miftah Zaini  
NIM : 2003321060  
Tanda Tangan :

Tanggal : 18 Agustus 2023



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkannya dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkannya dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun

LEMBAR PENGESAHAN  
TUGAS AKHIR

Tugas Akhir diajukan oleh:

Nama : Ibnu Miftah Zaini  
NIM : 2003321060  
Program Studi : D3 Elektronika Industri  
Judul Tugas Akhir : Control Open Loop Evaporator Fan dan Electric Expansion Valve (EEV) serta Monitoring Sensor NTC pada AC Bus Listrik Berbasis STMF407  
Sub Judul Tugas Akhir : Control Open Loop Stepper Motor Electric Expansion Valve (EEV) Menggunakan Komunikasi Canbus Berbasis STMF407

Telah diuji oleh tim penguji dalam Sidang Tugas Akhir pada 18, Agustus 2023 dan dinyatakan **LULUS**.

Pembimbing: Britantyo Wicaksono, S.Si., M.Eng  
NIP. 198404242018031001

Depok, 18 Agustus 2023

Disahkan oleh

Ketua Jurusan Teknik Elektro



Rika Nasyita Wardhani, S.T., M.T.

NIP. 197011142008122001



## KATA PENGANTAR

Puji syukur saya panjatkan kepada Tuhan Yang Maha Esa, karena atas berkat dan rahmat-Nya, maka tugas akhir dengan judul “*Control Open Loop pada Evaporator Fan dan Electrical Expansion Valve (EEV) serta Monitor NTC pada AC bus listrik berbasis STM32F407*” ini dapat diselesaikan dengan baik. Penulisan laporan Tugas Akhir ini dilakukan dalam rangka memenuhi salah satu syarat untuk mencapai gelar Diploma Tiga Politeknik, Penulis menyadari bahwa, tanpa bantuan dan bimbingan dari berbagai pihak, dari masa perkuliahan sampai pada penyusunan tugas akhir ini, sangatlah sulit bagi penulis untuk menyelesaikan tugas akhir ini. Oleh karena itu, penulis mengucapkan terima kasih kepada:

1. Rika Novita Wardhani, S.T., M.T selaku Ketua Jurusan Teknik Elektro.
2. Bapak Nuralam, S.T., M.T. selaku Ketua Program Studi Elektronika Industri.
3. Britantyo Wicaksono, ST, M.Eng. selaku dosen pembimbing yang telah menyediakan waktu, tenaga, dan pikiran untuk membimbing penulis dalam penyusunan laporan tugas akhir ini.
4. Orang tua dan keluarga penulis yang selalu memberikan dukungan, membantu secara finansial serta doa dalam menyelesaikan laporan tugas akhir ini.
5. PT. Arwito Indonesia yang telah mensupport dan mendanai alat yang sudah dibuat.
6. Rafli Hardi Fadilah dan Lavina Aura Muthmainah sebagai teman kelompok yang telah membantu penulis dalam penyusunan tugas akhir.
7. Teman-teman Angkatan 2020 yang telah memberikan dukungan.

Akhir kata, penulis berharap Tuhan Yang Maha Esa berkenan membalas segala kebaikan semua pihak yang telah membantu. Semoga laporan tugas akhir ini bermanfaat untuk masa depan nanti.

Depok, 13 Agustus 2023

Ibnu Miftah Zaini

### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkannya dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun



## CONTROL OPEN LOOP STEPPER MOTOR ELECTRICAL EXPANSION VALVE MENGUNAKAN KOMUNIKASI CANBUS BERBASIS STM32F407

### Abstrak

Transportasi umum pada saat ini banyak digunakan agar mengurangi kemacetan. Salah satunya ialah Bus. Pemakaian Bus konvensional sangat tidak ramah terhadap lingkungan. Bus listrik merupakan salah satu solusi yang sangat tepat dalam upaya penyelamatan lingkungan dan pengelolaan sumber alam secara berkelanjutan. Hal ini dimungkinkan karena teknologi pendukung kendaraan listrik telah berkembang sedemikian rupa sehingga telah berhasil dibuat kendaraan listrik yang handal, ekonomis, serta memiliki tingkat kenyamanan yang sama dengan kendaraan konvensional dan bahkan teknologi kendaraan listrik memiliki spesifikasi operasi yang lebih baik dibandingkan kendaraan konvensional. Salah satu tingkat kenyamanan yang didapat ialah pada sistem pendinginnya. Electric Expansion Valve (EEV) salah satu komponen yang dibutuhkan dalam pembuatan sistem pendingin pada bus listrik. Electric Expansion Valve (EEV) berada di tempat untuk mengatur aliran refrigerant dan memungkinkan lebih sedikit refrigerant mengalir melalui dan memperoleh uap yang sangat panas dengan sempurna. Berdasarkan pengujian dan analisa yang telah dilakukan dengan menggunakan penggaris busur derajat, pengecekan sudut derajat langsung dengan memberikan perintah step dari kontroler mendapatkan nilai rata-rata error sebesar 0,42% dan akurasi pengecekan sebesar 95%, pada hasil pengujian motor stepper menggunakan rotary encoder melalui head unit mendapatkan rata-rata error sebesar 4,7%, dan akurasi pengecekan sebesar 95%. Pada pengujian arus yang diukur pada motor stepper didapatkan rata-rata error sebesar 12,5%, dengan akurasi pengecekan sebesar 95%.

**Kata kunci :** Bus Listrik, Air Conditioner, Electric Expansion Valve (EEV)



## CONTROL OPEN LOOP STEPPER MOTOR ELECTRICAL EXPANSION VALVE USING CANBUS COMMUNICATION BASED ON STM32F407

### Abstract

Public transportation is currently widely used in order to reduce congestion. One of them is the Bus. The use of conventional buses is not very friendly to the environment. The electric bus is one of the most appropriate solutions in an effort to save the environment and manage natural resources in a sustainable manner. This is possible because electric vehicle support technology has been developed in such a way that electric vehicles have been successfully created that are reliable, economical, and have the same level of comfort as conventional vehicles and even electric vehicle technology has better operating specifications than conventional vehicles. One of the comfort levels you get is the cooling system. Electric Expansion Valve (EEV) is one of the components needed in making a cooling system on an electric bus. An Electric Expansion Valve (EEV) is in place to regulate refrigerant flow and allow less refrigerant to flow through and get the superheated vapor perfectly. Based on the tests and analyzes that have been carried out using a protractor, checking the direct angle by giving a step command from the controller gets an average error value of 0,42% and a checking accuracy of 95%, on the results of the stepper motor test using a rotary encoder through the head unit get an average error of 4,7%, and a checking accuracy of 95%. In testing the current measured on a stepper motor, an average error of 12.5% is obtained, with a checking accuracy of 95%.

**Keywords:** Electric Bus, Air Conditioner, Electric Expansion Valve (EEV)

POLITEKNIK  
NEGERI  
JAKARTA

#### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritis atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun



# DAFTAR ISI

HALAMAN PERNYATAAN ORISINALITAS .....	iii
LEMBAR PENGESAHAN TUGAS AKHIR.....	<b>Error! Bookmark not defined.</b>
KATA PENGANTAR .....	v
Abstrak .....	vi
DAFTAR ISI .....	viii
DAFTAR GAMBAR.....	x
DAFTAR TABEL .....	xi
DAFTAR LAMPIRAN .....	xii
BAB I.....	xli
PELDAHULUAN.....	xli
1.1 Latar Belakang .....	xli
1.2 Rumusan Masalah .....	xlii
1.3 Batasan Masalah.....	xlii
1.4 Tujuan.....	xlii
1.5 Luaran.....	xliii
BAB II.....	<b>Error! Bookmark not defined.</b>
TINJAUAN PUSTAKA.....	<b>Error! Bookmark not defined.</b>
2.1 Printed Circuit Board (PCB) .....	<b>Error! Bookmark not defined.</b>
2.2 CANBus .....	<b>Error! Bookmark not defined.</b>
2.3 Liquid Crystal Display (LCD) Nextion.....	<b>Error! Bookmark not defined.</b>
2.4 Electrical Expansion Valve (EEV).....	6
2.5 STM32F407 .....	<b>Error! Bookmark not defined.</b>
2.6 FUSION 360 .....	<b>Error! Bookmark not defined.</b>
2.7 STM32CubeIDE.....	<b>Error! Bookmark not defined.</b>
2.8 Power Supply .....	<b>Error! Bookmark not defined.</b>
2.9 Komponen SMD .....	<b>Error! Bookmark not defined.</b>
BAB III.....	14
PERENCANAAN DAN REALISASI .....	14
3.1 Perancangan Alat.....	<b>Error! Bookmark not defined.</b>
3.1.1 Perancangan Sistem .....	<b>Error! Bookmark not defined.</b>
3.1.2 Perancangan Program Sistem .....	<b>Error! Bookmark not defined.</b>
3.2 Realisasi Alat.....	19
3.2.1 Perancangan Rangkaian Elektronik Alat .....	20
3.2.2 Realisasi Pembuatan Rangkaian Board PCB.....	23

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :  
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.  
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta

2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun





Hak Cipta :

- 1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
- 2. Dilarang mengumumkannya dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun

3.2.3 Realisasi Pembuatan Casing 3D sebagai tempat PCB .....	25
3.2.4 Realisasi Program pada Control EEV ke Head Unit .....	<b>Error! Bookmark not defined.</b>
3.2.5 Realisasi Program pada Controller ke EEV .....	<b>Error! Bookmark not defined.</b>
3.2.6 Perancangan Mekanik.....	<b>Error! Bookmark not defined.</b>
BAHAGIAN IV .....	34
PEMBAHASAN .....	34
4.1 Pengujian Alat .....	34
4.1.1 Pengujian I Pengukuran Daya Pada Motor Stepper Terhadap Temperature Setting.....	34
4.1.2 Pengujian II Ketepatan dan Keakurasian Posisi Terhadap Temperature Setting.....	36
4.1.3 Pengujian III Ketepatan dan Keakurasian Posisi Sudut Terhadap Program.....	38
4.1.4 Analisis Data Pengujian.....	40
BAHAGIAN V .....	41
PELAKSANAAN .....	41
5.1 KESIMPULAN .....	41
5.2 SARAN .....	41
DAFTAR PUSTAKA .....	42
LAMPIRAN .....	xiv





## DAFTAR GAMBAR

Gambar 2. 1 PCB Head Unit.....	4
Gambar 2. 2. PCB Controller.....	4
Gambar 2. 3 Canbus.....	5
Gambar 2. 4 Sistem Kerja Canbus.....	6
Gambar 2. 5 LCD Nextion.....	6
Gambar 2. 6 Electric Expansion Valve.....	7
Gambar 2. 7 STM32F407.....	8
Gambar 2. 8 Fusion 360.....	8
Gambar 2. 9 STM32Cube IDE.....	9
Gambar 2. 10 Seri STM32 ARM Cortex-M tipe High Performance.....	10
Gambar 2. 11 Seri STM32 ARM Cortex-M tipe Mainstream.....	10
Gambar 2. 12 Seri STM32 ARM Cortex-M tipe Ultra Low Power.....	10
Gambar 2. 13 Seri STM32 ARM Cortex-M tipe Wireless.....	11
Gambar 2. 14 Power Supply 24V.....	11
Gambar 2. 15 Modul Step Down.....	11
Gambar 2. 16 Komponen SMD.....	13
Gambar 3. 1 Flowchart Keseluruhan.....	17
Gambar 3. 2 Diagram Blok Keseluruhan.....	18
Gambar 3. 3 Rangkaian Schematic Alat Keseluruhan.....	20
Gambar 3. 4 Rangkaian Schematic Electric Expansion Valve (EEV).....	21
Gambar 3. 5 Fitur New Schematic Pada Menu Utama Autodesk Fusion 360.....	21
Gambar 3. 6 Ikon Open Library manager Pada Autodesk Fusion360.....	22
Gambar 3. 7 Ikon Open Library Manager Pada Autodesk Fusion360.....	22
Gambar 3. 8 ikon Open File Board pada Fusion360.....	23
Gambar 3. 9 Rangkaian Schematic Board PCB Controller Tampak atas.....	24
Gambar 3. 10 Rangkaian Schematic Board PCB Controller Tampak Bawah.....	24
Gambar 3. 11 Rangkaian Schematic Board PCB Head Unit Tampak Atas.....	25
Gambar 3. 12 Rangkaian Schematic Board PCB Head Unit Tampak Bawah.....	25
Gambar 3. 13 Desain Box Head Unit.....	25
Gambar 3. 14 Desain Box Controller.....	26
Gambar 3. 15 Menu Plastic Pada Tab Autodesk Fusion360.....	26
Gambar 3. 16 Ikon Extrude Pada Tab Autodesk Fusion 360.....	27
Gambar 3. 17 Ikon Hole Pada Tab Autodesk Fusion360.....	27
Gambar 3. 18 Ikon Hell Pada Tab Autodesk Fusion360.....	28
Gambar 3. 19 Ikon Measure Pada Tab Autodesk Fusion360.....	28
Gambar 3. 20 Ikon Sketch Pada Tab Autodesk Fusion360.....	29
Gambar 3. 21 Pemrograman STM32CubeIDE.....	32
Gambar 3. 22 Pemrograman STM32CubeIDE.....	33
Gambar 3. 23 Bentuk Fisik Keseluruhan Alat.....	33

### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun



DAFTAR TABEL

Tabel 4. 1. Alat dan Bahan Pengujian ..... 34

Tabel 4. 2. Data Hasil Pengujian I ..... 35

Tabel 4. 3. Alat dan Bahan Pengujian..... 36

Tabel 4. 4. Data Hasil Pengujian II ..... 37

Tabel 4. 5. Alat dan Bahan Pengujian..... 38

Tabel 4. 6 Data Hasil Pengujian III..... 39



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun



### DAFTAR LAMPIRAN

Lampiran 1. Riwayat hidup..... xiv  
 Lampiran 2. Listing Pemrograman STM32F407 ..... xv  
 Lampiran 3. Dokumentasi Pick and Place ..... xxxvi  
 Lampiran 4. Dokumentasi Alat ..... xxxvii



**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun

**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkannya dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

## BAB I PENDAHULUAN

### 1.1 Latar Belakang

Adanya transportasi umum merupakan salah satu jalan keluar mengatasi kemacetan dan salah satu contoh transportasi umumnya yaitu bus. Bus yang tersedia dipasaran sekarang merupakan bus yang menggunakan bahan bakar fosil yang menyebabkan pemanasan global meningkat. Dengan beralih ke kendaraan listrik dapat mengurangi pemanasan global. Bus listrik merupakan salah satu contoh kendaraan listrik yang cocok dalam mengurangi risiko kemacetan sekaligus pemanasan global. (Merangin, 2018)

Untuk menciptakan masyarakat yang mayoritas menggunakan transportasi umum maka sudah sepantasnya transportasi yang ada dibuat senyaman mungkin. Salah satu faktor pembuat rasanyaman bagi manusia untuk menggunakan transportasi umum yaitu adanya sistem pendingin yang terdapat pada transportasi tersebut. Bus yang dilengkapi dengan sistem pengkondisian udara cenderung merupakan alternatif utama bagi para penumpang yang ingin menggunakan kendaraan angkutan umum seperti bus listrik. (Najamudin, 2018)

Komponen yang terdapat pada bus listrik yang saat ini digunakan masih menggunakan komponen buatan luar negeri. Untuk menekan angka TKDN maka pemerintah sedang gencar menggalakan pembuatan komponen yang terdapat pada bus listrik. Salah satu komponen yang akan dibuat untuk menekan angka TKDN pada bus listrik adalah sistem pendinginnya. Sistem pendingin pada bus listrik dapat menekan TKDN sebesar 31%.

Oleh karena itu, salah satu komponen yang terdapat pada sistem pendingin yang digunakan pada Bus Listrik ini ialah komponen *Electrical Expansion Valve* (EEV). EEV akan mengontrol aliran *refrigerant* ke *evaporator* dalam jumlah yang tepat. Peran EEV pada tugas akhir ini digantikan oleh motor *stepper*. Motor *stepper* pada kontroler dikendalikan oleh *rotary encode* yang terdapat pada *head unit*. Kontroler dan head unit saling terhubung melalui *canbus*, sekaligus sebagai sub judul pada tugas akhir ini.



**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

## 1.2 Rumusan Masalah

Berdasarkan latar belakang, dapat diperoleh perumusan masalah sebagai berikut:

- a. Bagaimana komunikasi antara *controller* dengan *head unit*?
- b. Bagaimana pengimplementasian motor stepper pada ac bus listrik?
- c. Bagaimana mengambil data nilai rotary encoder pada head unit dikirimkan melalui canbus ke *controller*, dan diolah menjadi perintah?
- d. Bagaimana merancang dan membuat sistem monitoring pada kontroller AC?
- e. Bagaimana membuat pemrograman untuk *Controller Air Conditioner* pada Bus Listrik?

## 1.3 Batasan Masalah

Adapun batasan masalah sebagai berikut :

1. Dalam perancangan ini, peran *Electrical Expansion Valve* (EEV) digantikan oleh motor *stepper*;
2. Untuk tegangan yang digunakan sebesar 12V;
3. Dalam perancangan ini pergerakan motor stepper dipengaruhi oleh besar kecilnya pengaturan suhu yang disetting pada head unit.

## 1.4 Tujuan

Berdasarkan rumusan masalah yang telah diutarakan, maka tujuan dari pembuatan Controller Air Conditioner (AC) pada Bus Listrik ialah sebagai berikut.

- a. Mengimplementasikan canbus sebagai alat komunikasi pada *Air Conditioner* (AC) Bus Listrik.
- b. Merancang dan membuat sistem perintah dengan mengambil data nilai dari rotary encoder, dikirim melalui *canbus*, dan diolah menjadi perintah pada AC Bus Listrik.
- c. Merancang dan membuat sistem monitoring melalui sebuah *controller* dan head unit pada AC Bus Listrik.
- d. Membuat pemrograman untuk *Controller AC* menggunakan *software* STM32Cube IDE.



## 1.5 Luaran

### Hak Cipta milik Politeknik Negeri Jakarta

#### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



## BAB V PENUTUP

### 1.6 KESIMPULAN

Berdasarkan Hasil Analisis dan Pengujian Control Openloop Pada Evaporator fan dan Electrical Expansion Valve (EEV) Serta Monitor NTC Pada AC Bus Listrik Berbasis STM32F407, maka dapat disimpulkan :

- Controller dan Head Unit mampu berkomunikasi melalui Canbus.
- Motor stepper dapat bergerak sesuai perintah yang diatur oleh rotary encoder pada head unit.
- Program yang dibuat pada stm32cubeide dapat beroperasi dengan baik.
- Untuk keakuratan motor stepper sudah memiliki keakuratan yang hampir mendekati dengan pesentase 85% tetapi masih ada selisih derajat yang diinginkan.
- Arus yang lewat masuk ke motor stepper stiap stepnya memiliki angka yang bervariasi.
- Untuk mengatur putaran motor stepper sesuai yang diinginkan maka untuk mengatur input yang terdapat pada pin step di DRV8825

### 1.7 SARAN

Saran untuk pada alat ini perlu ditingkatkan keakurasiannya, terutama pada sensor suhu NTC dan motor stepper. karena hasil data yang didapatkan belum menunjukkan angka yang akurat 100% masih terdapat error. Untuk display pada head unit sebaiknya dibuat lebih besar agar menarik dan desain box pada head unit ditambahkan tulisan pada setiap tombol dan rotary encoder.







## DAFTAR PUSTAKA

- Basri, Y., & Irfan, D. (2018). Komponen Elektronika. In *Sukabiina Press* (Vol. 53, Issue 9).
- Coker, C., Greene, E., Shao, J., Enclave, D., Tula, R., Marg, R., Jones, L., Hameiri, S., Cansu, E. E., Initiative, R., Maritime, C., Road, S., Çelik, A., Yaman, H., Turan, S., Kara, A., Kara, F., Zhu, B., Q. X., ... Tang, S. (2018). Software SRM32ARM. *Transcommunication*, 53(1), 1–8.
- Dwigista, C., Nataliana, D., & Anwari, S. (2022). Perancangan Dan Implementasi Printed Circuit Board ( Pcb ) Ramah Lingkungan Menggunakan Conductive Ink. *Jurnal POLEKTRO: Jurnal Power Elektronik*, 11(1), 31–35.
- Eni. (2018). STM32F407. *Angewandte Chemie International Edition*, 6(11), 951–952., Mi, 5–24.
- Fauzan, F., & Badarudin, A. (2022). Performansi Sistem AC Split Inverter Menggunakan Alat Ekspansi Pipa Kapiler dan Electronic Expansion Valve ( EEV ). *Prosiding The 13th Industrial Research Workshop and National Seminar*, 13, 13–14.
- Merandin. (2018). BUS LISTRIK. *Galang Tanjung*, 2504, 1–9.
- Najamudin. (2018). *Cara Menentukan Daya Yang Digunakan*. 12(1), 25–32.
- Setiawan, T. A., Juniani, A. I., Purnomo, D. A., & Rinanto, N. (2023). *HOW TO UTILIZE AUTODESK FUSION 360 THAT REINFORCES PRODUCT REDESIGN SIMULATION ?* 6, 48–54.
- TMicroelectronics. (2019). *Data Brief - Integrated development environment for STM32 products*. November.
- uhantono, D. (2019). Rancang Bangun Ac Power Supply Satu Fasa Step Down 230V / 12V Dengan Mengoptimalisasi Pengurangan Rugi-Rugi Dan Perbaikan Harmonisa. *Jurnal Logic*, 14(2), 75–81.
- Tiara Dewi, Muhammad Amir Masruhim, R. S. (2018). Desain Dan Implementasi Untuk Monitoring Dan Manajemen Energi Pada Charging Station Kendaraan Listrik Berbasis Can Bus. *Laboratorium Penelitian Dan Pengembangan FARMAKA TROPIS Fakultas Farmasi Universitas Mualawarman, Samarinda, Kalimantan Timur, April*, 5–24.

### Hak Cipta

### Hak Cipta Politeknik Negeri Jakarta

### Hak Cipta Politeknik Negeri Jakarta

### Hak Cipta Politeknik Negeri Jakarta

### Hak Cipta Politeknik Negeri Jakarta

2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

## LAMPIRAN

### Lampiran 1 Riwayat Hidup

#### DAFTAR RIWAYAT HIDUP PENULIS



Penulis bernama Ibnu Miftah Zaini, anak kedua dari dua bersaudara, lahir di Kebumen, 14 Desember 2000. Lulus dari SD Negeri 02 Petang Kedoya Utara tahun 2013, SMP Negeri 207 SSN Jakarta Barat tahun 2016, dan SMK Negeri 35 Jakarta tahun 2019. Gelar Diploma Tiga (D3) diperoleh pada tahun 2023 dari Jurusan Teknik Elektro, Program Studi Elektronika Industri, Politeknik Negeri Jakarta.

**POLITEKNIK  
NEGERI  
JAKARTA**



## Lampiran 2 Listing Program

### Listing Program

```

* @file      : main.c
* @brief     : Main program body
*****
* @attention
*
* Copyright (c) 2023 STMicroelectronics.
* All rights reserved.
*
* This software is licensed under terms that can be found in the LICENSE file
* in the root directory of this software component.
* If no LICENSE file comes with this software, it is provided AS-IS.
*
*****
*/
/* USER CODE END Header */
/* Includes -----*/
#include "main.h"
#include "cmsis_os.h"

/* Private includes -----*/
/* USER CODE BEGIN Includes */
// #include "datatypes.h"
/* USER CODE END Includes */

/* Private typedef -----*/
/* USER CODE BEGIN PTD */
#define UPDATE_CCR1(duty1) \
    TIM3->CR1 |= TIM_CR1_UDIS; \
    TIM3->CCR1 = duty1; \
    TIM3->CR1 &= ~TIM_CR1_UDIS;
/* USER CODE END PTD */

/* Private define -----*/
/* USER CODE BEGIN PD */
/* USER CODE END PD */

/* Private macro -----*/
/* USER CODE BEGIN PM */

/* USER CODE END PM */

/* Private variables -----*/
CAN_HandleTypeDef hcan1;

I2C_HandleTypeDef hi2c1;

```

#### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



## © Hak Cipta milik Politeknik Negeri Jakarta

### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
TIM_HandleTypeDef htim1;
TIM_HandleTypeDef htim3;
TIM_HandleTypeDef htim12;
DMA_HandleTypeDef hdma_tim3_ch1_trig;

UART_HandleTypeDef huart4;

osThreadId pwmHandle;
osThreadId stepperHandle;
osThreadId ntc03Handle;
osThreadId canbusHandle;
osThreadId myTask05Handle;
/* USER CODE BEGIN PV */
CAN_RxHeaderTypeDef rxHeader; //CAN Bus Transmit Header
CAN_TxHeaderTypeDef txHeader,txHeader1;

uint8_t canRX[8]; //CAN Bus Receive Buffer
CAN_FilterTypeDef canfil; //CAN Bus Filter
uint32_t canMailbox; //CAN Bus Mail box variable
uint8_t canRX_10[8];
uint8_t canRX_12[8];
uint16_t failsafe;
uint16_t txData;
uint8_t csend[8];
/* USER CODE END PV */

/* Private function prototypes -----*/
void SystemClock_Config(void);
static void MX_GPIO_Init(void);
static void MX_DMA_Init(void);
static void MX_TIM1_Init(void);
static void MX_TIM3_Init(void);
static void MX_CAN1_Init(void);
static void MX_I2C1_Init(void);
static void MX_UART4_Init(void);
static void MX_TIM12_Init(void);
void pwm_init(void const * argument);
void stepper_init(void const * argument);
void ntc03_init(void const * argument);
void canbus_init(void const * argument);
void StartTask05(void const * argument);

/* USER CODE BEGIN PFP */
char data;
void CAN_Transmit_SID(uint32_t id, const uint8_t *data, uint8_t len);
void CAN_Transmit_EID(uint32_t id, const uint8_t *data, uint8_t len);
void initCANFilter(void);
void Step(int stepVal ,int stateTarget);
long map(long x, long in_min, long in_max, long out_min, long out_max);
```



## © Hak Cipta milik Politeknik Negeri Jakarta

```
/* USER CODE END PFP */

/* Private user code -----*/
/* USER CODE BEGIN 0 */

#define DIR_PIN GPIO_PIN_10
#define DIR_PORT GPIOF
#define SLP_PIN GPIO_PIN_9
#define SLP_PORT GPIOF
#define RST_PIN GPIO_PIN_8
#define RST_PORT GPIOF
#define ENA_PIN GPIO_PIN_4
#define ENA_PORT GPIOF
#define STEP_PIN GPIO_PIN_14
#define STEP_PORT GPIOB
int stepDelay = 1000; // 1000us more delay means less speed

void microDelay (uint16_t delay)
{
    __HAL_TIM_SET_COUNTER(&htim1, 0);
    while (__HAL_TIM_GET_COUNTER(&htim1) < delay);
}

#define ADS1115_ADDRESS1 0x48
unsigned char ADSwrite1[6];
int16_t reading1;
float voltage1[4];
const float voltageConv1 = 6.114 / 32767.0;

#define ADS1115_ADDRESS 0x49
unsigned char ADSwrite[6];
int16_t reading;
float voltage[4];
const float voltageConv = 6.144 / 32767.0;

float resistance = 0;
float temperature = 0;
int
temperatureC = 0;

float resistance1 = 0;
float temperature1 = 0;
float temperatureC1 = 0;

float resistance2 = 0;
float temperature2 = 0;
float temperatureC2 = 0;

float resistance3 = 0;
float temperature3 = 0;
```

### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



## © Hak Cipta milik Politeknik Negeri Jakarta

### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
float temperatureC3 = 0;

float resistance4 = 0;
float temperature4 = 0;
float temperatureC4 = 0;

float resistance5 = 0;
float temperature5 = 0;
float temperatureC5 = 0;

//Variable data penampung CAN RX
uint8_t dataEncoderPos1 = 0;
uint8_t dataEncoderPos2 = 0;

int encoderTemp = 0;
int stateDir = 0;
int stepValue = 0;
int stepMotor = 0;
//Variable Stepper motor
int step=0,state=0,counter=0;
/* USER CODE END 0 */

/**
 * @brief The application entry point.
 * @retval int
 */
int main(void)
{
/* USER CODE BEGIN 1 */

/* USER CODE END 1 */

/* MCU Configuration-----*/

/* Reset of all peripherals, Initializes the Flash interface and the Systick. */
HAL_Init();

/* USER CODE BEGIN Init */

/* USER CODE END Init */

/* Configure the system clock */
SystemClock_Config();

/* USER CODE BEGIN SysInit */

/* USER CODE END SysInit */

/* Initialize all configured peripherals */
```



**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
MX_GPIO_Init();
MX_DMA_Init();
MX_TIM1_Init();
MX_TIM3_Init();
MX_CAN1_Init();
MX_I2C1_Init();
MX_UART4_Init();
MX_TIM12_Init();
/* USER CODE BEGIN 2 */
initCANFilter();

//Start Interface CAN1
HAL_CAN_Start(&hcan1);
// Activate the notification
HAL_CAN_ActivateNotification(&hcan1,CAN_IT_RX_FIFO0_MSG_PENDING);
// CAN1 untuk receive
__HAL_TIM_SET_COMPARE(&htim12,TIM_CHANNEL_1,128);
/* USER CODE END 2 */

/* USER CODE BEGIN RTOS_MUTEX */
/* add mutexes, ... */
/* USER CODE END RTOS_MUTEX */

/* USER CODE BEGIN RTOS_SEMAPHORES */
/* add semaphores, ... */
/* USER CODE END RTOS_SEMAPHORES */

/* USER CODE BEGIN RTOS_TIMERS */
/* start timers, add new ones, ... */
/* USER CODE END RTOS_TIMERS */

/* USER CODE BEGIN RTOS_QUEUES */
/* add queues, ... */
/* USER CODE END RTOS_QUEUES */

/* Create the thread(s) */
/* definition and creation of pwm */
osThreadDef(pwm, pwm_init, osPriorityNormal, 0, 512);
pwmHandle = osThreadCreate(osThread(pwm), NULL);

/* definition and creation of stepper */
osThreadDef(stepper, stepper_init, osPriorityNormal, 0, 1024);
stepperHandle = osThreadCreate(osThread(stepper), NULL);

/* definition and creation of ntc03 */
osThreadDef(ntc03, ntc03_init, osPriorityNormal, 0, 512);
ntc03Handle = osThreadCreate(osThread(ntc03), NULL);

/* definition and creation of canbus */
osThreadDef(canbus, canbus_init, osPriorityNormal, 0, 512);
```



- Hak Cipta :**
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
    - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
    - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
  2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
canbusHandle = osThreadCreate(osThread(canbus), NULL);

/* definition and creation of myTask05 */
osThreadDef(myTask05, StartTask05, osPriorityNormal, 0, 512);
myTask05Handle = osThreadCreate(osThread(myTask05), NULL);

/* USER CODE BEGIN RTOS_THREADS */
/* add threads, ... */
/* USER CODE END RTOS_THREADS */

/* Start scheduler */
osKernelStart();

/* We should never get here as control is now taken by the scheduler */
/* Infinite loop */
/* USER CODE BEGIN WHILE */
while (1)
{
    /* USER CODE END WHILE */

    /* USER CODE BEGIN 3 */
}
/* USER CODE END 3 */
}

/**
 * @brief System Clock Configuration
 * @retval None
 */
void SystemClock_Config(void)
{
    RCC_OscInitTypeDef RCC_OscInitStruct = {0};
    RCC_ClkInitTypeDef RCC_ClkInitStruct = {0};

    /** Configure the main internal regulator output voltage
     */
    __HAL_RCC_PWR_CLK_ENABLE();

    __HAL_PWR_VOLTAGESCALING_CONFIG(PWR_REGULATOR_VOLTAGE_SCALE1);

    /** Initializes the RCC Oscillators according to the specified parameters
     * in the RCC_OscInitTypeDef structure.
     */
    RCC_OscInitStruct.OscillatorType = RCC_OSCILLATORTYPE_HSI;
    RCC_OscInitStruct.HSIState = RCC_HSI_ON;
    RCC_OscInitStruct.HSICalibrationValue = RCC_HSICALIBRATION_DEFAULT;
    RCC_OscInitStruct.PLL.PLLState = RCC_PLL_ON;
    RCC_OscInitStruct.PLL.PLLSource = RCC_PLLSOURCE_HSI;
```





**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
RCC_OscInitStruct.PLL.PLLM = 8;
RCC_OscInitStruct.PLL.PLLN = 168;
RCC_OscInitStruct.PLL.PLLP = RCC_PLLP_DIV2;
RCC_OscInitStruct.PLL.PLLQ = 4;
if (HAL_RCC_OscConfig(&RCC_OscInitStruct) != HAL_OK)
{
    Error_Handler();
}

/** Initializes the CPU, AHB and APB buses clocks
*/
RCC_ClkInitStruct.ClockType =
RCC_CLOCKTYPE_HCLK|RCC_CLOCKTYPE_SYSCLK
|RCC_CLOCKTYPE_PCLK1|RCC_CLOCKTYPE_PCLK2;
RCC_ClkInitStruct.SYSCLKSource = RCC_SYSCLKSOURCE_PLLCLK;
RCC_ClkInitStruct.AHBCLKDivider = RCC_SYSCLK_DIV1;
RCC_ClkInitStruct.APB1CLKDivider = RCC_HCLK_DIV4;
RCC_ClkInitStruct.APB2CLKDivider = RCC_HCLK_DIV2;

if (HAL_RCC_ClockConfig(&RCC_ClkInitStruct, FLASH_LATENCY_5) !=
HAL_OK)
{
    Error_Handler();
}

/**
 * @brief CAN1 Initialization Function
 * @param None
 * @retval None
 */
static void MX_CAN1_Init(void)
{
    /* USER CODE BEGIN CAN1_Init 0 */

    /* USER CODE END CAN1_Init 0 */

    /* USER CODE BEGIN CAN1_Init 1 */

    /* USER CODE END CAN1_Init 1 */
    hcan1.Instance = CAN1;
    hcan1.Init.Prescaler = 70;
    hcan1.Init.Mode = CAN_MODE_NORMAL;
    hcan1.Init.SyncJumpWidth = CAN_SJW_1TQ;
    hcan1.Init.TimeSeg1 = CAN_BS1_2TQ;
    hcan1.Init.TimeSeg2 = CAN_BS2_3TQ;
    hcan1.Init.TimeTriggeredMode = DISABLE;
    hcan1.Init.AutoBusOff = DISABLE;
    hcan1.Init.AutoWakeUp = DISABLE;
```



- Hak Cipta :**
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
    - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
    - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
  2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
hcan1.Init.AutoRetransmission = DISABLE;
hcan1.Init.ReceiveFifoLocked = DISABLE;
hcan1.Init.TransmitFifoPriority = DISABLE;
if (HAL_CAN_Init(&hcan1) != HAL_OK)
{
    Error_Handler();
}
/* USER CODE BEGIN CAN1_Init 2 */

/* USER CODE END CAN1_Init 2 */
}
/**
 * @brief I2C1 Initialization Function
 * @param None
 * @retval None
 */
static void MX_I2C1_Init(void)
{
    /* USER CODE BEGIN I2C1_Init 0 */

    /* USER CODE END I2C1_Init 0 */

    /* USER CODE BEGIN I2C1_Init 1 */

    /* USER CODE END I2C1_Init 1 */
    hi2c1.Instance = I2C1;
    hi2c1.Init.ClockSpeed = 100000;
    hi2c1.Init.DutyCycle = I2C_DUTYCYCLE_2;
    hi2c1.Init.OwnAddress1 = 0;
    hi2c1.Init.AddressingMode = I2C_ADDRESSINGMODE_7BIT;
    hi2c1.Init.DualAddressMode = I2C_DUALADDRESS_DISABLE;
    hi2c1.Init.OwnAddress2 = 0;
    hi2c1.Init.GeneralCallMode = I2C_GENERALCALL_DISABLE;
    hi2c1.Init.NoStretchMode = I2C_NOSTRETCH_DISABLE;
    if (HAL_I2C_Init(&hi2c1) != HAL_OK)
    {
        Error_Handler();
    }
    /* USER CODE BEGIN I2C1_Init 2 */

    /* USER CODE END I2C1_Init 2 */

}
/**
 * @brief TIM1 Initialization Function
```



```
* @param None
* @retval None
*/
static void MX_TIM1_Init(void)
{
    /* USER CODE BEGIN TIM1_Init 0 */

    /* USER CODE END TIM1_Init 0 */

    TIM_ClockConfigTypeDef sClockSourceConfig = {0};
    TIM_MasterConfigTypeDef sMasterConfig = {0};

    /* USER CODE BEGIN TIM1_Init 1 */
    TIM_OC_InitTypeDef sConfigOC = {0};
    /* USER CODE END TIM1_Init 1 */
    htim1.Instance = TIM1;
    htim1.Init.Prescaler = 217;
    htim1.Init.CounterMode = TIM_COUNTERMODE_UP;
    htim1.Init.Period = 255;
    htim1.Init.ClockDivision = TIM_CLOCKDIVISION_DIV1;
    htim1.Init.RepetitionCounter = 0;
    htim1.Init.AutoReloadPreload = TIM_AUTORELOAD_PRELOAD_DISABLE;
    if (HAL_TIM_Base_Init(&htim1) != HAL_OK)
    {
        Error_Handler();
    }
    sClockSourceConfig.ClockSource = TIM_CLOCKSOURCE_INTERNAL;
    if (HAL_TIM_ConfigClockSource(&htim1, &sClockSourceConfig) != HAL_OK)
    {
        Error_Handler();
    }
    sMasterConfig.MasterOutputTrigger = TIM_TRGO_RESET;
    sMasterConfig.MasterSlaveMode = TIM_MASTERSLAVEMODE_DISABLE;
    if (HAL_TIMEx_MasterConfigSynchronization(&htim1, &sMasterConfig) !=
        HAL_OK)
    {
        Error_Handler();
    }
    /* USER CODE BEGIN TIM1_Init 2 */
    HAL_TIM_PWM_Init(&htim1);
    sConfigOC.OCMode = TIM_OCMODE_PWM1;
    sConfigOC.Pulse = 0;
    sConfigOC.OCpolarity = TIM_OCPOLARITY_HIGH;
    sConfigOC.OCFastMode = TIM_OCFAST_DISABLE;
    if (HAL_TIM_PWM_ConfigChannel(&htim1, &sConfigOC, TIM_CHANNEL_1)
        != HAL_OK){
        Error_Handler();
    }
}
```

#### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



```
HAL_TIM_MspPostInit(&htim1);
/* USER CODE END TIM1_Init 2 */
}

/**
 * @brief TIM3 Initialization Function
 * @param None
 * @retval None
 */
static void MX_TIM3_Init(void)
{
    /* USER CODE BEGIN TIM3_Init 0 */

    /* USER CODE END TIM3_Init 0 */

    TIM_ClockConfigTypeDef sClockSourceConfig = {0};
    TIM_MasterConfigTypeDef sMasterConfig = {0};
    TIM_OC_InitTypeDef sConfigOC = {0};

    /* USER CODE BEGIN TIM3_Init 1 */

    /* USER CODE END TIM3_Init 1 */
    htim3.Instance = TIM3;
    htim3.Init.Prescaler = 84-1;
    htim3.Init.CounterMode = TIM_COUNTERMODE_UP;
    htim3.Init.Period = 100-1;
    htim3.Init.ClockDivision = TIM_CLOCKDIVISION_DIV1;
    htim3.Init.AutoReloadPreload = TIM_AUTORELOAD_PRELOAD_DISABLE;
    if (HAL_TIM_Base_Init(&htim3) != HAL_OK)
    {
        Error_Handler();
    }
    sClockSourceConfig.ClockSource = TIM_CLOCKSOURCE_INTERNAL;
    if (HAL_TIM_ConfigClockSource(&htim3, &sClockSourceConfig) != HAL_OK)
    {
        Error_Handler();
    }
    if (HAL_TIM_PWM_Init(&htim3) != HAL_OK)
    {
        Error_Handler();
    }
    sMasterConfig.MasterOutputTrigger = TIM_TRGO_RESET;
    sMasterConfig.MasterSlaveMode = TIM_MASTERSLAVEMODE_DISABLE;
    if (HAL_TIMEx_MasterConfigSynchronization(&htim3, &sMasterConfig) !=
        HAL_OK)
    {
        Error_Handler();
    }
}
```

#### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



- Hak Cipta :**
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
    - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
    - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
  2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
sConfigOC.OCMode = TIM_OCMode_PWM1;
sConfigOC.Pulse = 0;
sConfigOC.OCpolarity = TIM_OCPolarity_HIGH;
sConfigOC.OCFastMode = TIM_OCFAST_DISABLE;
if (HAL_TIM_PWM_ConfigChannel(&htim3, &sConfigOC, TIM_CHANNEL_1)
    != HAL_OK)
{
    Error_Handler();
}
/* USER CODE BEGIN TIM3_Init 2 */

/* USER CODE END TIM3_Init 2 */
HAL_TIM_MspPostInit(&htim3);
}

/**
 * @brief TIM12 Initialization Function
 * @param None
 * @retval None
 */
static void MX_TIM12_Init(void)
{
    /* USER CODE BEGIN TIM12_Init 0 */

    /* USER CODE END TIM12_Init 0 */

    TIM_ClockConfigTypeDef sClockSourceConfig = {0};
    TIM_OC_InitTypeDef sConfigOC = {0};

    /* USER CODE BEGIN TIM12_Init 1 */

    /* USER CODE END TIM12_Init 1 */
    htim12.Instance = TIM12;
    htim12.Init.Prescaler = 217;
    htim12.Init.CounterMode = TIM_COUNTERMODE_UP;
    htim12.Init.Period = 255;
    htim12.Init.ClockDivision = TIM_CLOCKDIVISION_DIV1;
    htim12.Init.AutoReloadPreload = TIM_AUTORELOAD_PRELOAD_DISABLE;
    if (HAL_TIM_Base_Init(&htim12) != HAL_OK)
    {
        Error_Handler();
    }
    sClockSourceConfig.ClockSource = TIM_CLOCKSOURCE_INTERNAL;
    if (HAL_TIM_ConfigClockSource(&htim12, &sClockSourceConfig) != HAL_OK)
    {
        Error_Handler();
    }
    if (HAL_TIM_PWM_Init(&htim12) != HAL_OK)
```



## © Hak Cipta milik Politeknik Negeri Jakarta

### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
{
    Error_Handler();
}
sConfigOC.OCMode = TIM_OCMODE_PWM1;
sConfigOC.Pulse = 0;
sConfigOC.OCPolarity = TIM_OCPOLARITY_HIGH;
sConfigOC.OCFastMode = TIM_OCFAST_DISABLE;
if (HAL_TIM_PWM_ConfigChannel(&htim12, &sConfigOC, TIM_CHANNEL_1)
    != HAL_OK)
{
    Error_Handler();
}
/* USER CODE BEGIN TIM12_Init 2 */

/* USER CODE END TIM12_Init 2 */
HAL_TIM_MspPostInit(&htim12);
}

/**
 * @brief UART4 Initialization Function
 * @param None
 * @retval None
 */
static void MX_UART4_Init(void)
{
    /* USER CODE BEGIN UART4_Init 0 */

    /* USER CODE END UART4_Init 0 */

    /* USER CODE BEGIN UART4_Init 1 */

    /* USER CODE END UART4_Init 1 */
    huart4.Instance = UART4;
    huart4.Init.BaudRate = 115200;
    huart4.Init.WordLength = UART_WORDLENGTH_8B;
    huart4.Init.StopBits = UART_STOPBITS_1;
    huart4.Init.Parity = UART_PARITY_NONE;
    huart4.Init.Mode = UART_MODE_TX_RX;
    huart4.Init.HwFlowCtl = UART_HWCONTROL_NONE;
    huart4.Init.OverSampling = UART_OVERSAMPLING_16;
    if (HAL_UART_Init(&huart4) != HAL_OK)
    {
        Error_Handler();
    }
    /* USER CODE BEGIN UART4_Init 2 */

    /* USER CODE END UART4_Init 2 */
}
```



**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
}  
/**  
 * Enable DMA controller clock  
 */  
static void MX_DMA_Init(void)  
{  
    /* DMA controller clock enable */  
    __HAL_RCC_DMA1_CLK_ENABLE();  
  
    /* DMA interrupt init */  
    /* DMA1_Stream4_IRQn interrupt configuration */  
    HAL_NVIC_SetPriority(DMA1_Stream4_IRQn, 5, 0);  
    HAL_NVIC_EnableIRQ(DMA1_Stream4_IRQn);  
}  
  
/**  
 * @brief GPIO Initialization Function  
 * @param None  
 * @retval None  
 */  
static void MX_GPIO_Init(void)  
{  
    GPIO_InitTypeDef GPIO_InitStruct = {0};  
  
    /* GPIO Ports Clock Enable */  
    __HAL_RCC_GPIOF_CLK_ENABLE();  
    __HAL_RCC_GPIOH_CLK_ENABLE();  
    __HAL_RCC_GPIOA_CLK_ENABLE();  
    __HAL_RCC_GPIOB_CLK_ENABLE();  
    __HAL_RCC_GPIOC_CLK_ENABLE();  
  
    /*Configure GPIO pin Output Level */  
    HAL_GPIO_WritePin(GPIOF,  
GPIO_PIN_4|GPIO_PIN_8|GPIO_PIN_9|GPIO_PIN_10, GPIO_PIN_RESET);  
  
    /*Configure GPIO pin : PF0 */  
    GPIO_InitStruct.Pin = GPIO_PIN_0;  
    GPIO_InitStruct.Mode = GPIO_MODE_INPUT;  
    GPIO_InitStruct.Pull = GPIO_NOPULL;  
    HAL_GPIO_Init(GPIOF, &GPIO_InitStruct);  
  
    /*Configure GPIO pins : PF4 PF8 PF9 PF10 */  
    GPIO_InitStruct.Pin = GPIO_PIN_4|GPIO_PIN_8|GPIO_PIN_9|GPIO_PIN_10;  
    GPIO_InitStruct.Mode = GPIO_MODE_OUTPUT_PP;  
    GPIO_InitStruct.Pull = GPIO_PULLUP;  
    GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_LOW;  
    HAL_GPIO_Init(GPIOF, &GPIO_InitStruct);  
}
```



- Hak Cipta :**
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
    - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
    - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
  2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
    }
    /* USER CODE BEGIN 4 */
    //stepper motor
    void Step(int stepVal ,int stateVal)
    {
        step=stepVal;
        if(stateVal==0)
            HAL_GPIO_WritePin(DIR_PORT,DIR_PIN,GPIO_PIN_RESET);
        else
            HAL_GPIO_WritePin(DIR_PORT,DIR_PIN,GPIO_PIN_SET);
        state=1;
        HAL_TIM_PWM_Start_IT(&htim12,TIM_CHANNEL_1);
        while(1)
        {
            if(state==0)
                break;
            osDelay(1);
        }
    }
}

void HAL_TIM_PWM_PulseFinishedCallback(TIM_HandleTypeDef *htim)
{
    if(htim->Instance == TIM12)
    {
        counter++;
        if(counter == step)
        {
            HAL_TIM_PWM_Stop_IT(&htim12,TIM_CHANNEL_1);
            counter=0;
            state=0;
        }
    }
}

void HAL_CAN_RxFifo0MsgPendingCallback(CAN_HandleTypeDef *hcan1)
{
    //Receive menggunakan CAN1 kontroller lainnya
    if (HAL_CAN_GetRxMessage(hcan1, CAN_RX_FIFO0, &rxHeader,
canRX) != HAL_OK)//Get message dari FIFO0 CAN1
    {
        Error_Handler();
    }

    uint32_t EID = rxHeader.ExtId;
    uint8_t LengthBuffer = rxHeader.DLC;
}
```





Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
// additional filter
if(EID == 17){
    dataEncoderPos1 = canRX[1];
    dataEncoderPos2 = canRX[0];
}

failsafe = 3000;
}

void initCANFilter(void)
{
//
    canfil.FilterBank = 0;
    canfil.FilterMode = CAN_FILTERMODE_IDMASK;
    canfil.FilterScale = CAN_FILTERSCALE_32BIT;
    canfil.FilterIdHigh = 0x0000;
    canfil.FilterIdLow = 0x0000;
    canfil.FilterMaskIdHigh = 0x0000;
    canfil.FilterMaskIdLow = 0x0000;
    canfil.FilterFIFOAssignment = CAN_RX_FIFO0;
    canfil.FilterActivation = ENABLE;
//
    canfil.SlaveStartFilterBank = 14;
    HAL_CAN_ConfigFilter(&hcan1, &canfil);
}

//Fungsi untuk transmit frame CAN tipe Extended
void CAN_Transmit_EID(uint32_t id, const uint8_t *data, uint8_t len)
{
    if(len > 8){len = 8;}

    txHeader.IDE = CAN_ID_EXT;
    txHeader.ExtId = id;
    txHeader.RTR = CAN_RTR_DATA;
    txHeader.DLC = len;
//
    memcpy(TxData,data,len);

    //Transmit menggunakan CAN2 oleh headunit
    if(HAL_CAN_AddTxMessage(&hcan1, &txHeader, data, &canMailbox) !=
    HAL_OK){
        //Error
    }
}

//Fungsi untuk transmit frame CAN tipe Standar
void CAN_Transmit_SID(uint32_t id, const uint8_t *data, uint8_t len)
{
    if(len > 8){len = 8;}

    txHeader.IDE = CAN_ID_STD;
    txHeader.ExtId = id;
```



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
txHeader.RTR = CAN_RTR_DATA;
txHeader.DLC = len;
//
memcpy(TxData,data,len);

//Transmit menggunakan CAN2 oleh headunit
if(HAL_CAN_AddTxMessage(&hcan1, &txHeader, data, &canMailbox) !=
HAL_OK){
    //Error
}
}

long map(long x, long in_min, long in_max, long out_min, long out_max) {
    return (x - in_min) * (out_max - out_min) / (in_max - in_min) + out_min;
}
/* USER CODE END 4 */

/* USER CODE BEGIN Header_pwm_init */
/**
 * @brief Function implementing the pwm thread.
 * @param argument: Not used
 * @retval None
 */
/* USER CODE END Header_pwm_init */
void pwm_init(void const * argument)
{
    /* USER CODE BEGIN 5 */
    HAL_TIM_PWM_Start(&htim3, TIM_CHANNEL_1); //Start PWM from
timer 3 channel 1
    /* Infinite loop */
    for(;;)
    {
        if(dataEncoderPos1 == 0){
            UPDATE_CCR1(95); //95% ARR untuk menyalakan fan evap
5%
        }
        else if(dataEncoderPos1 > 0 && dataEncoderPos1 <= 35){
            UPDATE_CCR1(70); //70% ARR untuk menyalakan fan evap
30%
        }
        else if(dataEncoderPos1 > 35 && dataEncoderPos1 <= 70){
            UPDATE_CCR1(30); //30% ARR untuk menyalakan
fan evap 70%
        }else{
            UPDATE_CCR1(10); //10% ARR untuk menyalakan
fan evap 90%
        }
        osDelay(10);
    }
}
/* USER CODE END 5 */
}
```



**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
/* USER CODE BEGIN Header_stepper_init */
/**
 * @brief Function implementing the stepper thread.
 * @param argument: Not used
 * @retval None
 */
/* USER CODE END Header_stepper_init */
void stepper_init(void const * argument)
{
  /* USER CODE BEGIN stepper_init */
  HAL_GPIO_WritePin(GPIOF, GPIO_PIN_4, GPIO_PIN_RESET);
  HAL_GPIO_WritePin(GPIOF, GPIO_PIN_8, GPIO_PIN_SET);
  HAL_GPIO_WritePin(GPIOF, GPIO_PIN_9, GPIO_PIN_SET);

  /* Infinite loop */
  for(;;)
  {
    if(dataEncoderPos2 >= 30){
      dataEncoderPos2 = 29;
    }
    else{
      HAL_GPIO_WritePin(GPIOF,GPIO_PIN_4,GPIO_PIN_RESET);
      stepMotor = map(dataEncoderPos2,16,30,42,0);
      if(dataEncoderPos2 > encoderTemp){
        stateDir = 0;
      }else{
        stateDir = 1;
      }
      if((dataEncoderPos2 - encoderTemp) == 0){
      }else{
        Step(stepMotor,stateDir); // 200
      }
      stepMotor = 0;
      encoderTemp = dataEncoderPos2;
    }
    osDelay(100);
  }
  /* USER CODE END stepper_init */
}

/* USER CODE BEGIN Header_ntc03_init */
/**
 * @brief Function implementing the ntc03 thread.
 * @param argument: Not used
 * @retval None
 */
```



```

/* USER CODE END Header_ntc03_init */
void ntc03_init(void const * argument)
{
  /* USER CODE BEGIN ntc03_init */
  /* Infinite loop */
  for(;;)
  {
    for( int i = 0; i < 4; i++) {
      ADSwrite1[0] = 0x01;
      switch(i) {
        case(0):
          ADSwrite1[1] = 0xC1; //
          break;
        case(1):
          ADSwrite1[1] = 0xD1; //
          break;
        case(2):
          ADSwrite1[1] = 0xE1;
          break;
        case(3):
          ADSwrite1[1] = 0xF1;
          break;
      }
      ADSwrite1[2] = 0x83; // 10000011
      HAL_I2C_Master_Transmit(&hi2c1,
        ADS1115_ADDRESS1<<1, ADSwrite1, 3, 100);
      ADSwrite1[0] = 0x00;
      HAL_I2C_Master_Transmit(&hi2c1,
        ADS1115_ADDRESS1<<1, ADSwrite1, 1, 100);
      HAL_Delay(20);
      HAL_I2C_Master_Receive(&hi2c1,
        ADS1115_ADDRESS1<<1, ADSwrite1, 2, 100);
      reading1 = (ADSwrite1[0] << 8 |
        ADSwrite1[1]);
      if(reading1 < 0 ) {
        reading1 = 0;
      }
      voltage1[i] = (1/(reading1 *
        voltageConv1))+1;
      voltage1[0];
      + 1/(25+273.15);
      voltage1[1])/ voltage1[1];
      resistance = (5000 * (3.3 - voltage1[0]))/
      temperature = log(resistance/5000)/3950
      temperatureC = 1 / temperature - 273.15;
      resistance1 = (5000 * (3.3 -
      temperature1 =

```

**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



- Hak Cipta :**
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
    - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
    - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
  2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```

log(resistance1/5000)/3950 + 1/(25+273.15);
temperatureC1 = 1 / temperature1 -
273.15;

resistance2 = (5000 * (3.3 -
voltage1[2]))/ voltage1[2];
temperature2 =
log(resistance2/5000)/3950 + 1/(25+273.15);
temperatureC2 = 1 / temperature2 -
273.15;

resistance3 = (5000 * (3.3 -
voltage1[3]))/ voltage1[3];
temperature3 =
log(resistance3/5000)/3950 + 1/(25+273.15);
temperatureC3 = 1 / temperature3 -
273.15;
}
for( int a = 0; a < 4; a++) {
    ADSwrite[0] = 0x01;
    switch(a) {
        case(0):
            ADSwrite[1] =
            break;
        case(1):
            ADSwrite[1] =
            break;
        case(2):
            ADSwrite[1] =
            break;
        case(3):
            ADSwrite[1] =
            break;
    }
    ADSwrite[2] = 0x83; //
0xC1; // 11000001
0xD1; // 11010001
0xE1;
0xF1;
10000011
    HAL_I2C_Master_Transmit(&hi2c1, ADS1115_ADDRESS<<1, ADSwrite,
3, 100);
    ADSwrite[0] = 0x00;
    HAL_I2C_Master_Transmit(&hi2c1, ADS1115_ADDRESS<<1, ADSwrite,
1, 100);
    HAL_Delay(20);
    HAL_I2C_Master_Receive(&hi2c1, ADS1115_ADDRESS<<1, ADSwrite, 2,

```



## © Hak Cipta milik Politeknik Negeri Jakarta

### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```

100);
ADSwrite[1]);

voltageConv1))+1;
voltage[0])/ voltage[0];
log(resistance4/5000)/3950 + 1/(25+273.15);
- 273.15;
voltage[0])/ voltage[0];
log(resistance5/5000)/3950 + 1/(25+273.15);
- 273.15;
}
osDelay(1);
}
/* USER CODE END ntc03_init */
}

/* USER CODE BEGIN Header_canbus_init */
/**
 * @brief Function implementing the canbus thread.
 * @param argument: Not used
 * @retval None
 */
/* USER CODE END Header_canbus_init */
void canbus_init(void const * argument)
{
/* USER CODE BEGIN canbus_init */
/* Infinite loop */
for(;;)
{
    csend[0] = temperatureC;
    csend[1] = 0x02;
    csend[2] = 0x03;
    csend[3] = 0x04;
    csend[4] = 0x05;
    csend[5] = 0x06;
    csend[6] = 0x07;
    csend[7] = 0x08;
    CAN_Transmit_EID(103, csend, 8); //id nya 103 dengan data 8
    osDelay(200);

    reading = (ADSwrite[0] << 8 |
    if(reading < 0 ) {
    reading = 0;
    }
    voltage[a] = (1/(reading1 *
    resistance4 = (5000 * (3.3 -
    temperature4 =
    temperatureC4 = 1 / temperature4
    resistance5 = (5000 * (3.3 -
    temperature5 =
    temperatureC5 = 1 / temperature5
  
```



**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
}
/* USER CODE END canbus_init */
}

/* USER CODE BEGIN Header_StartTask05 */
/**
 * @brief Function implementing the myTask05 thread.
 * @param argument: Not used
 * @retval None
 */
/* USER CODE END Header_StartTask05 */
void StartTask05(void const * argument)
{
    /* USER CODE BEGIN StartTask05 */
    /* Infinite loop */
    for(;;)
    {
        osDelay(1);
    }
    /* USER CODE END StartTask05 */
}

/**
 * @brief This function is executed in case of error occurrence.
 * @retval None
 */
void Error_Handler(void)
{
    /* USER CODE BEGIN Error_Handler_Debug */
    /* User can add his own implementation to report the HAL error return state */
    __disable_irq();

    while (1)
    {
    }
    /* USER CODE END Error_Handler_Debug */
}

#ifdef USE_FULL_ASSERT
/**
 * @brief Reports the name of the source file and the source line number
 * where the assert_param error has occurred.
 * @param file: pointer to the source file name
 * @param line: assert_param error line source number
 * @retval None
 */
void assert_failed(uint8_t *file, uint32_t line)
{
    /* USER CODE BEGIN 6 */
    /* User can add his own implementation to report the file name and line number,

```



## © Hak Cipta milik Politeknik Negeri Jakarta

```
ex: printf("Wrong parameters value: file %s on line %d\r\n", file, line) */  
/* USER CODE END 6 */  
}  
#endif /* USE_FULL_ASSERT */
```

### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



### Lampiran 3 Dokumentasi Pick and Place





## © Hak Cipta milik Politeknik Negeri Jakarta

### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



© Hak Cipta milik Politeknik Negeri Jakarta



**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta