Infotekmesin
Vol.14, No.02, Juli 2023
p-ISSN: 2087-1627, e-ISSN: 2685-9858
DOI: 10.35970/infotekmesin.v14i2.1924, pp.354-361

Sistem Komunikasi Mikrokontroler dan PLC Berbasis Komunikasi Serial *Host Link* dan Protokol *C-Command* RS232

Dewi Rizani Ruwahida[1*], Isa Rachman[2], Hendro Agus Widodo[3], Ryan Yudha Adhitya[4] , Yuda Irawan[5]
[1, 2]Program Studi Teknik Otomasi, Politeknik Perkapalan Negeri Surabaya
[1,2] Jl. Teknik Kimia, Keputih, Kec. Sukolilo, Surabaya, Jawa Timur 60111, Indonesia
E-mail: @dewirizani@student.ppns.ac.id[1], isarachman@ppns.ac.id[2], hendro.aw@ppns.ac.id[3], ryanyudhaadhitya@ppns.ac.id[4], yudairawan@student.ppns.ac.id[5]

Info Naskah:
Naskah masuk: 29 Mei 2023
Direvisi: 13 Juli 2023
Diterima: 13 Juli 2023

Abstrak

Komunikasi mikrokontroler dan PLC menjadi salah satu pengembangan teknologi yang penting untuk ditingkatkan. Kebutuhan komunikasi dua perangkat ini salah satunya untuk memudahkan proses input data atau informasi dari sensor yang berperan untuk mendeteksi atau mengukur beberapa parameter tertentu. Tidak semua sensor dapat langsung terbaca oleh PLC, beberapa sensor memiliki protokol komunikasi atau library khusus yang tidak didukung secara langsung oleh PLC. Mikrokontroler sering digunakan sebagai perantara antara sensor dan PLC. Mikrokontroler dapat membaca data sensor yang kemudian mengubahnya menjadi format yang dapat dimengerti oleh PLC. Mikrokontroler dapat melakukan pemrosesan datar seperti konversi hingga penilaian status untuk memberikan data yang lebih relevan dan kemudian mengirimkan data tersebut pada PLC melalui protokol komunikasi yang kompatibel. Dalam penelitian ini protokol yang digunakan adalah komunikasi serial Host Link dengan menggunakan metode C-mode Command. Hasil pengujian komunikasi yang telah dilakukan menggunakan 10 macam data sampel didapatkan nilai yang akurat dengan perolehan persentase error sebesar 0%.

Keywords:
communication;
microcontroller;
programmable logic controller;

Abstract

The communication between microcontrollers and PLC has become an important technology development that needs to be enhanced. One of the reasons for the communication between these two devices is to facilitate the input process of data or information from sensors, which are responsible for detecting or measuring specific parameters. Not all sensors can be directly read by PLC, as some sensors have their own communication protocols or specific libraries that are not directly supported by PLC. Microcontrollers are often used as intermediaries between sensors and PLC. Microcontrollers can read sensor data and then convert it into a format that can be understood by PLC. Microcontrollers can perform flat processing tasks such as conversion and status evaluation to provide more relevant data, and then send this data to the PLC through a compatible communication protocol. In this study, the communication protocol used is the Host Link serial communication using the C-mode Command method. The communication testing conducted using 10 different data samples resulted in accurate values with an error percentage of 0%.

***Penulis korespondensi:**
Dewi Rizani Ruwahida
E-mail: dewirizani@student.ppns.ac.id

1. Pendahuluan

Protokol komunikasi memegang peranan penting dalam mengatur komunikasi antara perangkat yang dirancang dengan cara yang berbeda sesuai dengan persyaratan sistem. Protokol ini mengikuti aturan khusus yang disepakati antara perangkat untuk mencapai komunikasi yang efektif dan berhasil[1]. Komunikasi dapat dibedakan menjadi dua yaitu serial dan paralel. Komunikasi serial digunakan untuk mentransmisikan data jarak jauh yang hanya memerlukan satu saluran untuk berkomunikasi dan mentransfer data. Sedangkan transmisi paralel memerlukan banyak saluran dan digunakan untuk jarak yang lebih pendek[2]. Komunikasi serial digunakan untuk mentransfer satu bit pada satu waktu melalui media tertentu, sedangkan dalam komunikasi paralel, satu blok data ditransfer pada saat yang sama dengan masing-masing bit membutuhkan saluran terpisah[3]. Komunikasi serial akan mengambil beberapa bentuk sehubungan dengan jenis mode transmisi data. Mode transmisi data diklasifikasikan sebagai *simplex*, *half dupleks*, dan *full-dupleks*[4].

UART(*Universal Asynchronous Receiver/Transmitter*) adalah *microchip* dengan pemrograman yang mengontrol antarmuka komputer ke perangkat serial yang terpasang[5]. Protokol komunikasi ini digunakan untuk mengirim dan menerima data secara serial antara perangkat elektronik. UART pada dasarnya memiliki dua bagian, *transmitter* dan *receiver*. *Transmitter* atau pemancar mengambil data paralel dari komputer dan mengirimkan data secara serial. *Receiver* atau penerima mengambil data secara serial dari perangkat dan mengirimkannya ke komputer secara paralel [6].TxD adalah sinyal data yang dikirimkan dan RxD adalah sinyal data serial yang diterima [7].

Penelitian komunikasi *half dupleks* secara serial yang pernah diterapkan yaitu menggunakan RS485 dimana pengiriman dan penerimaan data dapat dilakukan pada jalur yang sama atau bergantian. Kelebihan dari pengaplikasian ini adalah komunikasi dapat dilakukan dengan jarak yang jauh serta memiliki toleransi tinggi terhadap noise, namun RS485 memiliki konfigurasi yang cukup rumit dan kompatibilitas yang rendah [8].

Pada penelitian ini mengkomunikasikan antara mikrokontroler jenis Arduino Nano melalui *software* Arduino IDE dengan PLC Omron CJ1M CPU21 menggunakan komunikasi serial RS232 dan kemudian menampilkan hasil komunikasinya berupa pengiriman data yang ditampilkan pada *interface* visual studio. RS232 adalah standar yang umum digunakan dalam komunikasi serial yang bekerja dengan baik dalam jarak pendek sekitar 15 meter di mana memadai untuk banyak aplikasi industri dan proses kontrol. Untuk menghubungkan kedua *hardware* tersebut terdapat beberapa mode komunikasi serial yang dapat digunakan oleh PLC Omron, seperti Host Link, Sysmac Gateway dan FINS (*Factory Interface Network Service*)[9].

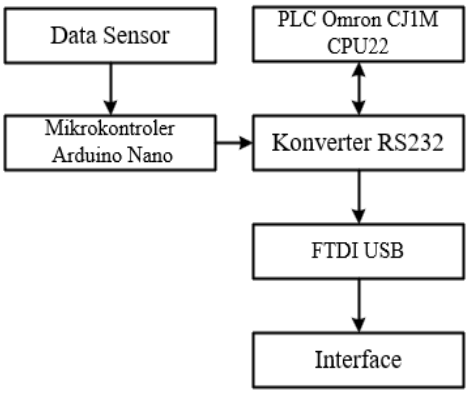
Mode Host Link memungkinkan perangkat *host* untuk berkomunikasi dengan peralatan OMRON yang mendukung protokol. Ini termasuk semua PLC seri C dan CV dan MMI seri NT (*Man Machine Interfaces*)[10]. Dalam mode *Host Link*, PLC berfungsi sebagai "*slave*" atau perangkat yang menerima permintaan dari "*master*" atau perangkat yang mengirim permintaan. Komunikasi antara *master* dan *slave* dilakukan melalui kabel serial RS-232 melalui pin Rx maupun Tx. Host link dapat digunakan untuk mengirim perintah C-mode Host link Command dari *host* untuk membaca/menulis memori I/O, kontrol mode pengoperasian, dll [11].

Dalam penelitian ini menggunaan C-Mode Command sebagai metode komunikasi antara perangkat host dan PLC. Ketika menggunakan Mode Host Link, perintah (*command*) dikirim dari perangkat *host* ke PLC menggunakan format pesan yang sesuai dengan C-Mode Command. PLC akan menerima perintah tersebut, menjalankan operasi yang diminta, dan mengirimkan respons (*response*) kembali ke perangkat *host*.

2. Metode

2.1 Konsep

Pada Gambar 1 merupakan gambaran singkat dari konsep penelitian ini. Sistem yang akan dirancang dan dibangun adalah *monitoring* mengenai pengiriman data melalui *interface* visual studio. PLC modular CJ1M CPU21 yang terintegrasi dengan mikrokontroler sebagai kontrol sekaligus pengendali kinerja dan kondisi dari perangkat-perangkat yang beroperasi.



Gambar 1. Konsep Penelitian

Pada Gambar 1 merupakan penjelasan dari masing-masing perangkat yang digunakan. Data Sensor atau *Input* merupakan satu data atau lebih yang akan dikirimkan. Data ini dapat diperoleh dari nilai sensor maupun *input* nilai yang ingin dikirimkan. Mikrokontroler Arduino Nano merupakan salah satu jenis papan pengembangan mikrokontroler yang berbasis Arduino menggunakan mikrokontroler ATMega328 atau ATMega168, yang memiliki arsitektur AVR. Arduino dapat digunakan untuk berkomunikasi dengan komputer, papan Arduino lain, atau mikrokontroler lainnya. Mikrokontroler ATmega328P menyediakan komunikasi serial UART TTL (5V) yang dapat dilakukan menggunakan pin digital 0 (Rx) dan pin digital 1 (Tx)[12]. Mikrokontroler ini memiliki kecepatan *clock* 16 MHz dan RAM serta EEPROM yang cukup untuk kebanyakan aplikasi [13]. Masing-masing dari 14 pin digital pada Nano dapat digunakan sebagai *input* atau *output*, menggunakan pinMode(), digitalWrite(), dan fungsi digitalRead(). Digunakan untuk menerima (RX) dan mengirimkan (TX) data serial TTL. Pin ini terhubung pada pin yang sesuai dari chip FTDI USB-to-TTL Serial[14].

Konverter RS232 secara *hardware* adalah MAXRS232 yang merupakan *driver*/penerima ganda yang termasuk generator tegangan kapasitif untuk memasok Level tegangan TIA/EIA-232-F dari 5-V tunggal Pasokan. Setiap penerima mengonversi input TIA/EIA-232-F ke level 5-V TTL/CMOS. Penerima ini memiliki ambang tipikal 1,3 V, histeresis tipikal 0,5 V, dan dapat menerima masukan ±30-V. Setiap driver mengkonversi Level masukan TTL/CMOS ke level TIA/EIA-232-F[15]. Dalam penelitian ini MAXRS232 digunakan sebagai penghubung atau alat bantu komunikasi mikrokontroler to PLC maupun sebaliknya. RS-232 adalah standar komunikasi serial yang didefinisikan sebagai antarmuka antara perangkat terminal data DTE (*data terminal equipment*) dan perangkat komunikasi data DCE (*data communications equipment*) menggunakan pertukaran data biner secara serial[16]. *Programmable Logic Controller*, singkatnya PLC merupakan suatu bentuk khusus pengendalian berbasis mikroprossesor yang memanfaatkan memori yang dapat di program untuk menyimpan intruksi-intruksi dan untuk menerapkan (*counting*) dan aritmetika guna mengendalikan mesin-mesin dan proses-proses[17].

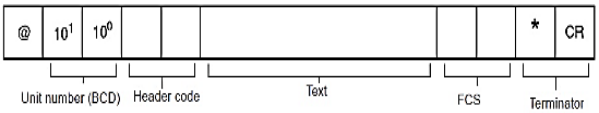
FTDI USB merupakan USB *programmer all-in-one* berbasis *chip* CH341 yang dapat digunakan untuk mengkonversi antarmuka USB pada PC menjadi I²C, TTL, USB, SPI, UART dan lain sebagainya, sehingga PC tersebut dapat berkomunikasi dan melakukan pemrograman terhadap mikrokontroler ataupun *development tool* lainnya[18]. *Interface* Visual Studio mengacu pada antarmuka pengembangan terpadu (*Integrated Development Environment*/IDE) yang digunakan dalam perangkat lunak Microsoft Visual Studio. Visual Studio adalah suite perangkat lunak yang digunakan oleh para pengembang perangkat lunak untuk membuat aplikasi desktop, aplikasi web, aplikasi seluler, game, dan banyak lagi[19].

2.2 C-Command Mode

Dalam komunikasi PLC OMRON menggunakan metode C-Command, terdapat dua macam *frame format* yaitu *command frame format* dan *response frame format*. *command frame format* berfungsi mengirimkan sebuah perintah atau kode sedangkan *response frame format* berfungsi untuk menerima atau merespon sebuah perintah atau kode yang telah dikirim. *command frame format* dan *response frame format* memiliki format penulisan yang berbeda yang mana tergatung dari segi kebutuhan. Ketentuan penulisan tersebut dapat dilihat pada *datasheet* Omron. Dalam penelitian ini kebutuhan komunikasi tersebut hanya untuk mengirimkan data saja.

2.2.1 Command frame format Status Change

Command frame format Status Change adalah salah satu format *frame* yang digunakan untuk mengendalikan dan memonitor status PLC Omron melalui perangkat lain yang terhubung dengannya melalui komunikasi serial. *Command frame format Status Change* dapat dilihat pada Gambar 2.



Gambar 2. *Command frame format Status Change*[9]

Gambar 2 merupakan *command frame format* status change yang berfungsi untuk mengubah status unit pada PLC. Dalam penelitian ini untuk melakukan komunikasi dengan PLC, status unit pada PLC adalah monitor. Mode monitor sangat berguna karena memungkinkan pengguna untuk melihat data yang diterima dan dikirim oleh PLC. Dalam mode monitor, pengguna dapat melihat nilai variabel-variabel program, status input/output, serta informasi lain yang diperlukan untuk memastikan bahwa PLC sedang beroperasi dengan benar. Pada Tabel 1 merupakan penjelasan dari *Command frame format Status Change*. *Command frame format status change* ini sering digunakan selama pengembangan program PLC, debugging, atau pemecahan masalah (troubleshooting) untuk memeriksa apakah program PLC sedang berjalan dengan benar atau tidak.

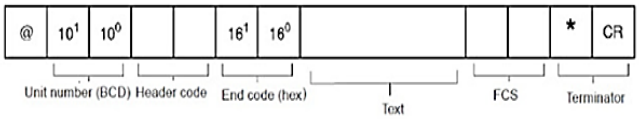
Tabel 1. Keterangan *Command frame format Status Change*

Tanda	Keterangan
@	Harus selalu disertakan pada awal *frame* respon.
Unit number	Bernilai BCD 0-31 untuk setiap unit host link PLC
Header code.	Merupakan kode perintah ditentukan dengan dua karakter
Teks	Berisikan set parameter sesuai kode perintah
FCS	Hitung dari 2 karakter FCS pada host komputer
Terminator	Merupakan dua karakter yang harus disertakan pada akhir *frame* untuk menunjukkan akhir perintah

2.2.2 Response frame format Status Change

Format *frame* respons (*response frame*) pada komunikasi serial berisi respons atau tanggapan (response) terhadap perintah status change. Format *frame* respons (*response frame*) dapat dilihat pada Gambar 3. Pada Gambar 3 merupakan *Response frame format Status Change.* Menunjukkan respon PLC apakah komunikasi berhasil atau tidak.

Pada Tabel 2 merupakan penjelasan dari *Response frame format Status Change.* Dalam format ini terdapat *end code* yang berisikan kode informasi keberhasilan komunikasi pada PLC.



Gambar 3. *Response frame format Status Change*[9]

Tabel 2. Keterangan *Response frame format Status Change*

Tanda	Keterangan
@	Harus selalu disertakan pada awal *frame* respon.
Unit number	Bernilai BCD 0-31 untuk setiap unit host link PLC
Header code.	Merupakan kode perintah ditentukan dengan dua karakter
End code	Merupakan hasil eksekusi perintah
Teks	Berisikan set parameter sesuai kode perintah (hanya jika ada perintah membaca data memori)
FCS	Hitung dari 2 karakter FCS pada host komputer
Terminator	Merupakan dua karakter yang harus disertakan pada akhir *frame* untuk menunjukkan akhir perintah

2.3 FCS (*Frame Check Sequence*)

FCS adalah hasil konversi 8-bit data ke 2 digit karakter ASCII (*American Standard Code for Information Interchange*) dimana merupakan kode berupa angka yang merepresentasikan karekter-karakter, baik huruf, angka, maupun simbol yang digunakan oleh komputer[20]. Ke 8-bit data merupakan hasil dari *exclusive* OR secara berurut (*sequence*) karakter pertama hingga karakter terakhir pada sebuah *frame*.

Contoh perhitungan *Command Frame Format*:

- Jika input biner PLC yaitu 0000 0003. Hitung data TX yang harus dikirim pada memori PLC sesuai dengan *command format*?

Tahap 1: Menghitung kode hexa untuk input Teks

Tabel 3. Tahap 1 Perhitungan *Command Frame Format*

Biner	Kode Hexa
0000 0003	0 3

Pada Tabel 3 merupakan data input text 03 yang dituliskan dalam bentuk biner dan hexa. Maka kode yang dikirimkan pada PLC berbentuk @00WD03(FCS)*

Tahap 2: Mencari kode ASCII untuk input FCS

Tabel 4. Tahap 2 Perhitungan *Command Frame Format*

Input	Kode ASCII
@	40
0	30
0	30
W	57
D	44
0	33
3	30

Pada Tabel 4 merupakan tahap dua dari proses perhitungan FCS. Dalam tahap ini masing-masing karakter kode dinyatakan dengan nilai ASCII.

Tahap 3: Perhitungan FCS

Pada Tabel 5 merupakan perhitungan final dari FCS. Dengan demikian maka data yang dikirimkan pada input Text 03 sama dengan kode @00WD0350*. Jika data yang dikirim benar yaitu @00WD0350* maka respons yang diberikan adalah @00WD000350* artinya normal *completion* atau penyelesaian berjalan normal. Jika data yang dikirim salah yaitu @00SC0359* maka respons yang diberikan adalah @00WD1303FCS*↵ artinya FCS Error.

Tabel 5. Tahap 3 Perhitungan *Command Frame Format*

Code	Bil. Biner	Bil. Biner
4 0	0100	0000
3 0	0011	0000
3 0	0011	0000
5 7	0101	0111
4 4	0100	0100
3 3	0011	0011
3 0	0011	0000
xOR	**0101**	**0000**
Bil. Hexa	**5**	**0**

Tabel 6. Tabel *End code* (Hex)

End code (Hex)	Contents
00	Normal competion
13	FCS error
14	Format error
15	Entry number data error
18	*Frame* length error
21	Not execitable due to CPU Unit CPU error

Pada Tabel 6 merupakan beberapa contoh dari *end code* yang akan muncul pada respon komunikasi dimana menunjukkan beberapa artian, salah satunya adalah *end code* "00" yang mengartikan bahwa komunikasi normal tanpa error.

3. Hasil dan Pembahasan

Berdasarkan Gambar 4 Komunikasi yang digunakan merupakan jenis komunikasi serial dengan pengaturan kecepatan transmisi data (*baud rate)* sebesar 9600 bit per detik dan format data menjadi 7 bit data, *even parity*, dan 2 bit *stop* bit yang mana sesuai dengan konfigurasi pada PLC yang digunakan. Selanjutnya adalah perintah untuk mengubah mode monitor pada PLC menggunakan *Command frame format* "@00SC0252*".

```
Serial.begin(9600, SERIAL_7E2);
Serial.print("@00SC0252*");
```

Gambar 4. Program Komunikasi Mikro to PLC

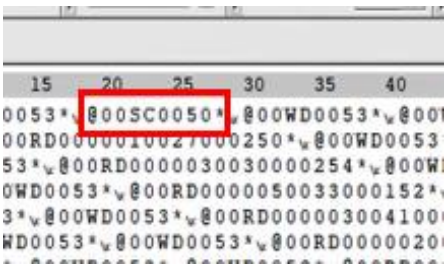
Tabel 7. Keterangan *Command frame format* "@00SC0252*"

Kode	Keterangan
@	Karakter yang harus disertakan pada awal
00	Unit number binery code untuk setiap unit Host Link
SC	*Header code* perintah untuk mengubah status unit PLC
02	Teks perintah pemilihan mode monitor pada PLC
52	Hasil perhitungan FCS
*	Terminator

Pada Tabel 7 *command frame format status unit* mode monitor yang digunakan untuk melihat nilai dari variabel-variabel program dan mengecek status PLC. Mode ini sering digunakan selama pengembangan program PLC, *debugging*, atau pemecahan masalah (*troubleshooting*) untuk memeriksa apakah program PLC sedang berjalan dengan benar atau tidak. Ketika melakukan komunikasi dengan PLC, mode monitor sangat berguna karena memungkinkan pengguna untuk melihat data yang diterima dan dikirim oleh PLC. Dalam mode *monitor*, pengguna dapat melihat nilai variabel-variabel program, status *input/output*, serta informasi lain yang diperlukan untuk memastikan bahwa PLC sedang beroperasi dengan benar.

Pada Gambar 5 merupakan data respon PLC dengan format @00SC0050* yang menunjukkan bahwa data tersebut sudah diterima oleh PLC. Penjelasan kode tersebut adalah seperti pada Tabel 8. Pada Tabel 8 merupakan respon *frame* format yang memiliki *end code* 00 di mana menunjukkan bahwa komunikasi yang dilakukan telah berhasil dan sukses.

Selanjutnya terdapat program fungsi yang bernama WD0 yang digunakan untuk menulis data ke dalam perangkat PLC melalui protokol komunikasi Host Link. Fungsi ini menerima sebuah argumen berupa string yang merepresentasikan alamat memori pada PLC yang akan ditulis datanya. Dalam keseluruhan fungsi WD0 tersebut, program akan menuliskan data ke alamat memori pada PLC sesuai dengan nilai string1 yang diberikan. Fungsi ini akan mengembalikan nilai berupa pesan yang siap dikirimkan ke PLC untuk menulis data pada alamat memori yang telah ditentukan. Selanjutnya adalah pengujian program dengan mengirim data sensor berupa nilai 3.



Gambar 5. Respon *Frame* Format "@00SC0252*"

Tabel 8. Keterangan Respon *Frame* Format "@00SC0252*"

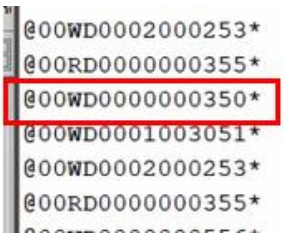
Kode	Keterangan
@	Karakter yang harus disertakan pada awal perintah
00	Unit number binery code untuk setiap unit Host Link
SC	*Header code* perintah untuk mengubah status unit PLC
00	*End code*; *normal completion* atau penyelesaian berjalan normal
50	Perhitungan FCS
*	Terminator

```
dat1 = 3;
dat11 = String (dat1);
WD0(dat11);
delay(500);
```

Gambar 6. Pengiriman Data Sensor

Pemrograman pada Gambar 6 diatas merupakan contoh pengiriman data sensor yang dimisalkan dengan variabel dat1 yang mana nilainya adalah 3. Variabel ini sebagai data atau nilai yang akan dikirimkan melalui komunikasi serial. Pada baris kedua, nilai variabel "dat1" diubah menjadi string menggunakan fungsi "String()". Hasil konversi ini akan disimpan dalam variabel "dat11". Penulisan fungsi "String()" di sini berfungsi untuk mengubah nilai numerik ke dalam bentuk string yang nantinya akan dikirimkan melalui kabel serial. Pada baris ketiga, fungsi "WD0" dipanggil dengan parameter "dat11". Fungsi "WD0" merupakan fungsi yang digunakan untuk mengirimkan pesan melalui komunikasi serial. Pada baris keempat, diberikan jeda selama 500 milidetik menggunakan fungsi "delay()". Jeda tersebut dibutuhkan agar pesan yang dikirim tidak terlalu cepat dan memberi waktu bagi perangkat penerima untuk menangkap dan memproses pesan.

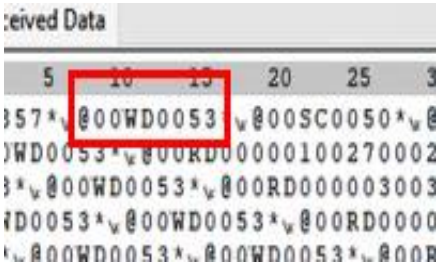
Pada Gambar 7 merupakan data yang akan dikirimkan dengan format @00WD0000000350* dengan keterangan seperti pada Tabel 9. Pada Tabel 9 merupakan format *frame* data yang akan dikirimkan ke memori PLC. Dalam format ini kode yang dikirim adalah WD yang berarti *Write data*. Pada penelitian ini salah satu memori acuannya adalah D0 dengan nilai data kirim adalah 3.



Gambar 7. Format Data Kirim WD

Tabel 9. Keterangan Format Data Kirim WD

Kode	Keterangan
@	Karakter yang harus disertakan pada awal perintah
00	Unit number binery code untuk setiap unit Host Link
WD	Header code "WD" adalah kode yang digunakan untuk mengirimkan data (*write data*) ke perangkat PLC melalui protokol komunikasi yang telah ditentukan. Biasanya, header code "WD" akan diikuti oleh alamat memori yang akan ditulis datanya dan nilai data yang akan ditulis.
0000	Alamat memori yang akan ditulis datanya
0003	Nilai data yang ingin dikirim
50	Perhitungan FCS
*	Terminator

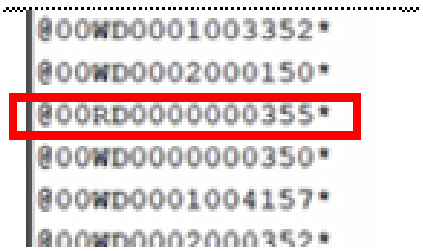


Gambar 8. Respon Data Terima WD

Pada Gambar 8 merupakan data respon PLC dengan format @00WD0053* yang menunjukkan bahwa data tersebut sudah diterima oleh PLC. Penjelasan kode tersebut adalah pada Tabel 10. Tabel 10 menunjukkan respon *frame* dari data terima WD. Dalam respon ini *end code* yang didapatkan adalah 00 yang berarti komunikasi penerimaan data telah sukses atau berhasil. Selanjutnya adalah sebuah fungsi yang bernama RD yang digunakan untuk membaca data memori dari perangkat PLC melalui protokol komunikasi Host Link. Fungsi ini menerima sebuah argumen berupa string yang merepresentasikan alamat memori pada PLC yang akan dibaca datanya. Dalam keseluruhan fungsi RD tersebut, program akan membaca alamat memori pada PLC sesuai dengan nilai string1 yang diberikan. Fungsi ini akan mengembalikan nilai berupa pesan yang siap dikirimkan ke PLC untuk membaca data dari alamat memori yang telah ditentukan.

Tabel 10. Keterangan Respon Data Terima WD

Kode	Keterangan
@	Karakter yang harus disertakan pada awal perintah
00	Unit number binery code untuk setiap unit Host Link
WD	Header code "WD" adalah kode yang digunakan untuk mengirimkan data (*write data*) ke perangkat PLC melalui protokol komunikasi yang telah ditentukan.
00	*End code*; *normal completion* atau penyelesaian berjalan normal
53	Perhitungan FCS
*	Terminator



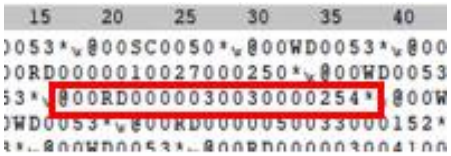
Gambar 9. Format Data Kirim RD

Tampilan pada serial monitor pada Gambar 9 diatas menunjukkan bahwa program membaca tiga buah alamat memori pada PLC yaitu D0, D1, dan D2 dikarenakan pada penelitian ini banyaknya variabel yang dikirimkan adalah sebanyak tiga dengan nilai yang berbeda. Dalam Tabel 11 berisi format data kirim RD yang mengartikan *Read data*. Pada penelitian ini membutuhkan tiga memori PLC. Dengan demikian perintah permintaan memori telah dikirimkan sebanyak 3 memori.

Tabel 11. Keterangan Format Data Kirim RD

Kode	Keterangan
@	Karakter yang harus disertakan pada awal perintah
00	Unit number binery code untuk setiap unit Host Link
RD	Header code "RD" adalah kode yang digunakan untuk membaca data (*read data*) dari perangkat PLC melalui protokol komunikasi yang telah ditentukan. Biasanya, header code "RD" akan diikuti oleh alamat memori yang akan dibaca dan jumlah data yang akan dibaca..
0000	Begin memory; alamat memori awal yang dibaca
0003	Nilai dari jumlah memori yang dibutuhkan (ada 3 memori)
55	Perhitungan FCS
*	Terminator

Pada Gambar 10 merupakan data respon PLC dengan format yang menunjukkan bahwa data kirim untuk masing-masing memori tersebut sudah diterima oleh PLC. Penjelasan kode tersebut adalah pada Tabel 12. Pada Tabel 12 adalah respon dari data terima RD dengan *end code* yang didapatkan adalah 00 yang berarti komunikasi penerimaan data telah sukses atau berhasil. Selanjutnya adalah kode pemrograman untuk penerimaan data yang diproses pada visual studio.



Gambar 10. Respon Data Terima RD

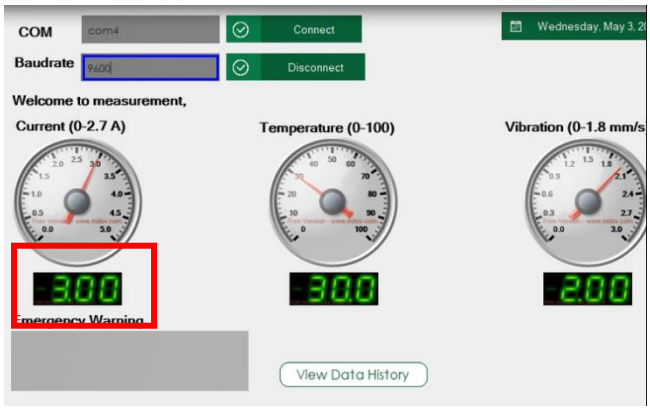
Tabel 12. Keterangan Respon Data Terima RD

Kode	Keterangan
@	Karakter yang harus disertakan pada awal perintah
00	Unit number binery code untuk setiap unit Host Link
RD	Header code "RD" adalah kode yang digunakan untuk membaca data (*read data*) dari perangkat PLC melalui protokol komunikasi yang telah ditentukan. Biasanya, header code "RD" akan diikuti oleh alamat memori yang akan dibaca dan jumlah data yang akan dibaca.
00	*End code*; ; *normal completion* atau penyelesaian berjalan normal
0003	Nilai data pertama yang diterima; untuk memori D0
0030	Nilai data kedua yang diterima; untuk memori D1
0002	Nilai data ketiga yang diterima; untuk memori D2
54	Perhitungan FCS
*	Terminator

```
Private Sub SerialPort1_DataReceived;
AngularGauge1.Value = Mid(data_masuk, 8, 4)
SegmentGauge1.Value = Mid(data_masuk, 8, 4)
AngularGauge2.Value = Mid(data_masuk, 12, 4)
SegmentGauge2.Value = Mid(data_masuk, 12, 4)
AngularGauge3.Value = Mid(data_masuk, 16, 4)
SegmentGauge3.Value = Mid(data_masuk, 16, 4)
```

Gambar 11. Program Parsing Data Visual Studio

Pada Gambar 11 adalah pemrograman kode singkat Visual Studio yang berfungsi untuk mengambil data dari port serial perangkat lain, seperti PLC atau mikrokontroler yang kemudian memisahkan data-data sesuai dengan kebutuhan dan menampilkan data tersebut pada *interface*. Gambar 12 merupakan uji pengiriman data PLC dengan PC yaitu *interface* visual studio dimana dapat dilihat bahwa antara data kirim dengan data yang diterima dan ditampilkan adalah sama. Penjelasan format data dapat dilihat pada pembahasan sebelumnya. Pada bagian yang ditandai merupakan data sensor yang bernilai 3, sesuai dengan pengujian program yang telah dijelaskan sebelumnya



Gambar 12. Uji Pengiriman Data

Tabel 13. Uji Hasil Pengiriman Data

No.	Serial Monitor	PLC	Visual Studio	Error %
1.	1	1	1	0
2.	**3**	**3**	**3**	0
3.	5	5	5	0
4.	3	3	3	0
5.	2	2	2	0
6.	4	4	4	0
7.	5	5	5	0
8.	5	5	5	0
9.	5	5	5	0
10.	1	1	1	0
Rata-rata persentase *error*				0

Pada Tabel 13 merupakan pengujian pengiriman 10 data sampel untuk mengetahui keakuratan dari komunikasi antara mikrokontroler dengan PLC. Berdasarkan hasil persentase *error* pada tabel dapat diketahui bahwa komunikasi pengiriman data berhasil dengan baik dengan persentase error sebesar 0%.

4. Kesimpulan

Pada penelitian ini integrasi antara mikrokontroler dengan PLC melalui komunikasi serial Host Link c-command mode RS232 dapat dikatakan berhasil dan akurat dengan perolehan persentase error sebesar 0%. RS232 mendukung komunikasi serial hingga jarak 50 kaki (sekitar 15 meter) tanpa perlu penggunaan perangkat tambahan seperti repeater. Ini memudahkan pengaturan sistem komunikasi dalam jarak yang terbatas. RS232 jugs memiliki format yang relatif sederhana dan mudah dipahami. Pengaturan dan konfigurasi perangkat RS232 juga cenderung lebih mudah dibandingkan dengan teknologi komunikasi yang lebih kompleks. Komunikasi mikrokontroler dan PLC Omron merupakan riset pertama yang dilakukan. Beberapa keandalan yang ditemukan dalam penelitian ini diharapkan dapat menjadi pedoman pada riset-riset selanjutnya.

Ucapan Terimakasih

Terimakasih kepada Politeknik Perkapalan Negeri Surabaya yang telah memfasilitasi sarana serta prasarana dalam melakukan penelitian ini.

Daftar Pustaka

[1] E. Peña and M. G. Legaspi, "Uart: A hardware communication protocol understanding universal asynchronous receiver/transmitter," *Visit Analog*, vol. 54, no. 4, 2020.

[2] D. S. Dawoud and P. Dawoud, *Serial Communication Protocols and Standards*. CRC Press, 2022.

[3] A. Gupta and A. Gupta, "UART communication," *IoT hacker's Handb. a Pract. Guid. to hacking Internet things*, pp. 59–80, 2019.

[4] H. N. Y. Pwint, T. Kywe, and T. T. E. Aung, "PC AND PIC BASED ELECTRONIC DEVICES CONTROLLER USING SERIAL COMMUNICATION," *Int. J. All Res. Writings*, vol. 2, no. 3, pp. 129–133, 2019.

[5] M. Poorani and R. Kurunjimalar, "Design implementation of UART and SPI in single FGPA," in *2016 10th International Conference on Intelligent Systems and Control (ISCO)*, 2016, pp. 1–5.

[6] A. K. Gupta, A. Raman, N. Kumar, and R. Ranjan, "Design and implementation of high-speed universal asynchronous receiver and transmitter (UART)," in *2020 7th International Conference on Signal Processing and Integrated Networks (SPIN)*, 2020, pp. 295–300.

[7] U. Nanda and S. K. Pattnaik, "Universal asynchronous receiver and transmitter (uart)," in *2016 3rd international conference on advanced computing and communication systems (ICACCS)*, 2016, vol. 1, pp. 1–5.

[8] A. Mardiyanto, "Monitoring System for Organic Fertilizer Plant Controller Using Solar Energy," in *IOP Conference Series: Materials Science and Engineering*, 2019, vol. 536, no. 1, p. 12050.

[9] OMRON, "Serial Communications Boards and Serial Communications Units," 2009. https://assets.omron.eu/downloads/manual/en/w336_cs1_cj1_serial_communications_boards_operation_manual_en.pdf

[10] O. Corporation, "Sample Programs for Host Link Commands," 2009. https://www.myomron.com/index.php?action=kb&article=80

[11] O. Corporation, "Host Link Communications," 2009. https://paginas.fe.up.pt/~pfs/recursos/plcs/omron/cqm1/sbc_manual/sec4.pdf

[12] A. Zakaria and A. Prihantara, "Pemanfaatan Radio Frequency Identification Mifare RC522 dan Arduino Sebagai Media Validasi Kehadiran Mahasiswa," *J. Infotekmesin*, vol. 11, no. 01, 2020.

[13] A. Kurniawan, *Arduino Nano A Hands-on Guide for Beginner*. PE press, 2019.

[14] A. Nano, "Arduino Nano," *A MOBICON Co.*, 2018.

[15] T. Instrument, "MAX232x Dual EIA-232 Drivers/Receivers," *Texas Instrum.*, 2014.

[16] D. S. Dawoud and P. Dawoud, *Serial Communication Protocols and Standards RS232/485, UART/USART, SPI,*

USB, INSTEON, Wi-Fi and WiMAX. River Publishers, 2020.
[17] M. Naim, *Buku Ajar Sistem Kontrol dan Kelistrikan Mesin*. Penerbit NEM, 2021.
[18] C. Hafiz, "RANCANG BANGUN SYSTEM MONITORING RUMAH BERBASIS MIKROKONTROLER DAN TELEGRAM." Institut Teknologi Telkom Jakarta, 2022.
[19] M. Wali, *Membangun Aplikasi Windows dengan Visual Basic. NET 2015 Teori dan Praktikum: Indonesia*, vol. 1. KITA Publisher, 2017.
[20] A. Elmogy, Y. Bouteraa, R. Alshabanat, and W. Alghaslan, "A New Cryptography Algorithm Based on ASCII Code," in *2019 19th International Conference on Sciences and Techniques of Automatic Control and Computer Engineering (STA)*, 2019, pp. 626–631.