
PENGEMBANGAN FIRMWARE PADA SUB CONTROLLER ROBOT SEPAK BOLA HUMANOID MENGGUNAKAN PROTOKOL DYNAMIXEL 2.0

Agung Pangestu¹, Maulana Aziz Assuja², Saniati Saniati³, Try Susanto⁴

^{1,2,3,4}Universitas Teknokrat Indonesia

Jl. ZA. Pagar Alam No.9 -11, Labuhan Ratu, Bandar Lampung, Indonesia 35132

agung.pangestu12@gmail.com

Abstract

Firmware is a software layer between the hardware and the operating system with the main objective of initializing the hardware so that the operating system and drivers can configure the hardware. This study aims to develop the existing firmware on the Krakatau Sub-Controller using the Dynamixel 2.0 protocol. The Dynamixel protocol is a UART (Universal Asynchronous Receiver-Transmitter) half-duplex communication protocol with 8-bit data length, 1-stop bit, and non-parity. The stages of the research that will be carried out are designing and implementing the Krakatau Sub Controller Board Version 2.0. The next step is that the rules in the Dynamixel 2.0 Protocol are translated into the C++ programming language and embedded in the Krakatau Sub Controller as firmware. The test scenarios carried out are functional testing by sending the bit value of each instruction and non-functional testing by testing the latency aspect. The firmware on the Krakatau Sub-Controller was successfully made with the latency time required for the Ping test of 12 μ S, read test of 114 μ S, write test of 11 μ S, Reg Write test of 7 μ S, Action test of 9 μ S, Factory Reset test of 14 μ S, Reboot test is 16 μ S, Clear test is 3 μ S, Sync Read test is 6 μ S, Bulk Read test is 17 μ S.

Keywords: *Dynamixel 2.0, Robotic, Sub-Controller, Krakatau FC, Humanoid.*

Abstrak

Firmware merupakan lapisan perangkat lunak antara perangkat keras dan sistem operasi dengan tujuan utama menginisialisasi perangkat keras, sehingga sistem operasi dan driver dapat mengkonfigurasi perangkat keras. Penelitian ini bertujuan mengembangkan firmware yang ada pada Krakatau Sub-Controller menggunakan protokol Dynamixel 2.0. Protokol Dynamixel adalah protokol komunikasi half-duplex UART (Universal Asynchronous Receiver-Transmitter) dengan panjang 8-bit data, 1 stop bit, dan none paritas. Tahapan penelitian yang akan dilakukan yaitu merancang dan mengimplementasikan board Krakatau Sub Controller Versi 2.0. Langkah selanjutnya yaitu aturan yang ada pada Protokol Dynamixel 2.0 diterjemahkan dalam bahasa pemrograman C++ dan ditanam ke Krakatau Sub Controller sebagai firmware. Skenario pengujian yang dilakukan yaitu pengujian fungsional dengan mengirim nilai bit dari tiap-tiap instruction dan pengujian non fungsional dengan menguji aspek latensi. Firmware pada Krakatau Sub-Controller berhasil dibuat dengan latensi waktu yang dibutuhkan untuk pengujian Ping yaitu 12 μ S, pengujian Read yaitu 114 μ S, pengujian Write yaitu 11 μ S, pengujian Reg Write yaitu 7 μ S, pengujian Action yaitu 9 μ S, pengujian Factory Reset yaitu 14 μ S, pengujian Reboot yaitu 16 μ S, pengujian Clear yaitu 3 μ S, pengujian Sync Read yaitu 6 μ S, Pengujian Bulk Read yaitu 17 μ S.

Kata kunci: *Dynamixel 2.0, Robotic, Sub-Controller, Krakatau FC, Humanoid*

1. PENDAHULUAN

Kontes Robot Sepak Bola Indonesia Humanoid (KRSBI-H) adalah kompetisi robot sepak bola berbentuk humanoid (manusia) dengan misi mencetak gol ke gawang lawan. RSBH Krakatau FC secara keseluruhan mengadopsi platform dari robot DARwIn-OP (Dynamic Anthropomorphic Robot with Intelligence-Open Platform). DARwIn-OP merupakan platform robot humanoid yang dikembangkan oleh Robotis, perusahaan asal Korea [1]. Robot Sepak Bola Humanoid Krakatau FC menggunakan sensor kamera sebagai deteksi citra pada lingkungan. Citra yang ditangkap oleh kamera akan dikirim ke main-controller menggunakan jalur USB. Pengolahan citra dilakukan oleh main-controller sesuai dengan algoritma pengolahan citra yang digunakan dan akan menghasilkan sekumpulan paket instruksi. Paket instruksi tersebut akan dikirim ke alamat yang sudah ditentukan berdasarkan identitas unik (ID) melalui sub-controller [2].

Krakatau Sub-Controller (KSC) adalah rangkaian elektronika terintegrasi yang berfungsi sebagai firmware. Menurut Sun et al. [3], firmware merupakan lapisan perangkat lunak antara perangkat keras dan sistem operasi dengan tujuan utama menginisialisasi perangkat keras, sehingga sistem operasi dan driver dapat mengkonfigurasi perangkat keras. Firmware pada sub-controller ditujukan untuk mengendalikan komunikasi data dari/menju main-controller. Permasalahan pada KSC versi 1.3 yaitu terdapat jalur komunikasi yang tidak efektif antara main-controller dengan aktuator. Data yang akan menuju ke aktuator harus melewati mikrocontroller yang ada pada KSC. Hal tersebut mengakibatkan terjadinya antrian eksekusi data. Antrian eksekusi data yang menuju aktuator tersebut menyebabkan robot mengalami keterlambatan gerak. Berdasarkan permasalahan tersebut, tim Krakatau FC melakukan pengembangan KSC versi 2.0 dengan perubahan pada jalur komunikasi data. Pada KSC versi 2.0, data dari main-controller yang menuju aktuator memiliki jalur sendiri dan tidak melewati mikrocontroller. Firmware yang ada pada KSC versi 1.3 tidak dapat diterapkan pada KSC versi 2.0 karena memiliki jalur komunikasi yang berbeda. Oleh karena itu, penulis berinisiatif melakukan penelitian dengan judul Pengembangan Firmware Pada Sub-Controller Robot Sepak Bola Humanoid Krakatau Fc Menggunakan Protokol Dynamixel 2.0.

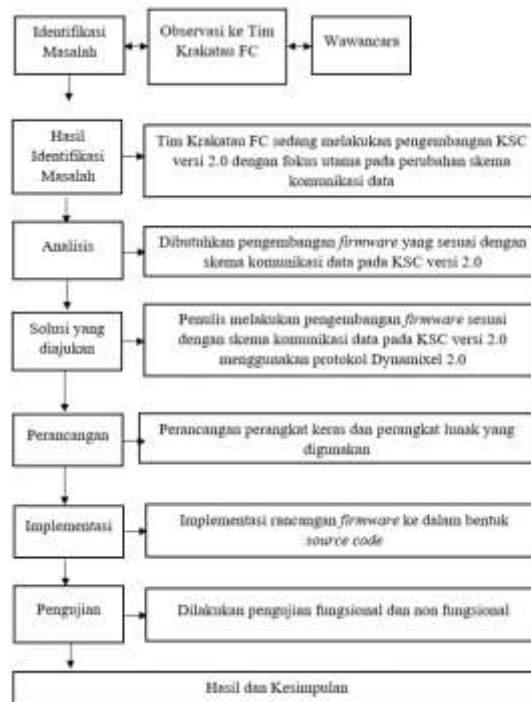
Protokol Dynamixel sendiri adalah protokol komunikasi half-duplex UART (Universal Asynchronous Receiver-Transmitter) dengan panjang 8-bit data, 1 stop bit, dan none paritas. Cara kerja dari protokol komunikasi Dynamixel adalah menerima data dalam bentuk binary. Main-controller dan aktuator Dynamixel saling berkomunikasi dengan mengirim dan menerima data yang disebut Paket. Paket memiliki dua jenis yaitu paket instruksi dan paket status. Paket instruksi adalah perintah yang dikirim oleh main-controller untuk mengontrol Dynamixel. Paket status adalah respon yang dikirim oleh Dynamixel ke main-controller. Aturan protokol komunikasi tersebut menggunakan sekumpulan paket instruksi [4]. Pada penelitian ini, akan dibuat firmware yang sesuai dengan jalur komunikasi pada KSC versi 2.0 menggunakan protokol Dynamixel 2.0 seperti yang telah dilakukan pada penelitian [5]. Penelitian ini juga dilakukan berdasarkan penelitian lain terkait protokol Dynamixel yang telah dilakukan oleh Bestmann, et.al [6], Hunter, et al [7], Yusuf, et al [8], Ha, et al [9], dan Fabre, et al [10].

Penelitian serupa pernah dilakukan oleh [11] yang berjudul Rancang Bangun Sistem Tertanam pada Humanoid Robot Soccer Tim Gladiatos UI. Kesimpulan dari penelitian tersebut yaitu Putra dkk telah berhasil melakukan pengembangan driver yang digunakan untuk komunikasi antara main-controller dan aktuator. Pada penelitian tersebut, metode yang digunakan adalah protokol Dynamixel 1.0. Penelitian lain juga pernah dilakukan oleh [12], dengan judul Rancang Bangun Robot Devata New Generation. Kesimpulan dari penelitian tersebut yaitu Susanto dkk berhasil membuat firmware yang dapat digunakan untuk komunikasi antara main-controller dan sub-controller. Firmware tersebut dibuat

menggunakan protokol Dynamixel. Main-controller yang digunakan yaitu Raspberry Pi dan sub-controller yang digunakan yaitu CM-530. Berdasarkan latar belakang tersebut, penelitian ini dilakukan dengan tujuan yaitu menerapkan protokol Dynamixel 2.0 untuk pengembangan firmware KSC Versi 2.0

2. METODOLOGI PENELITIAN

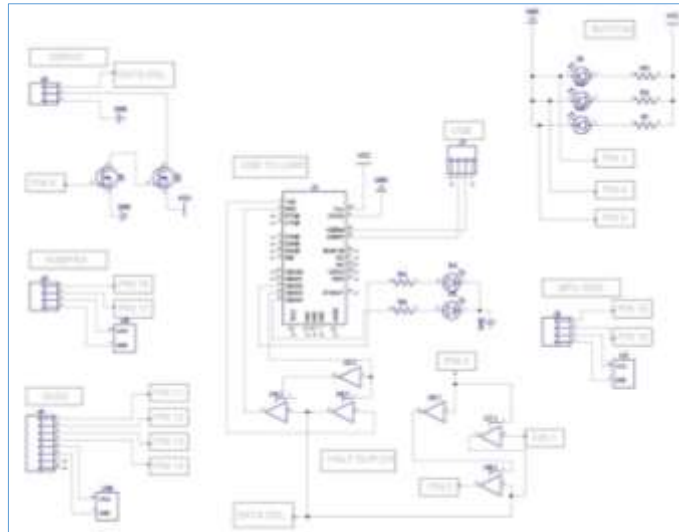
Objek dari penelitian ini yaitu Krakatau Sub-Controller (KSC) yang ada pada robot sepak bola humanoid Krakatau FC. Tahap awal penelitian adalah identifikasi masalah yang ada pada Krakatau Sub-Controller untuk kemudian dilakukan penelitian terhadap Krakatau Sub-Controller. Penelitian ini menerapkan langkah-langkah seperti pada gambar 1. Sedangkan kebutuhan perangkat keras dan perangkat lunak yang akan digunakan dapat dilihat pada table 1.



Gambar 1 Tahapan Penelitian

Tabel 1 Alat dan Bahan

Spesifikasi Hardware		
No	Spesifikasi	Deskripsi
1	Processor	Intel(R) Core(TM) i3-4005U CPU @ 1.70 GHz
2	RAM	4GB
3	Seagate	Seagate ST500LT012-1DG14 500 GB
4	Monitor	Intel(R) HD Graphics Family
5	Aktuator MX-64T	Pengujian penerima perintah
6	KSC Versi 2.0	Sub-Controller
7	Intel Nuc	Main-Controller



Gambar 3 Skematik KSC versi 2.0

Tabel 2 Penggunaan Pin Teensy 4.0

Pin Teensy 4.0	Terhubung Ke	Pin
0	IC 74HC126	Y3
1	IC 74HC126	A4
2	IC 74LS04	6A
3	Button Eksternal 1	-
4	Button Eksternal 2	-
5	Button Eksternal 2	-
6	TR BSS 138	G (Gate)
10	OLED	CS
11	OLED	D IN
12	OLED	D OUT
13	OLED	DC
15	POWER 16 Volt	-
16	Kompas Eksternal	SCL
17	Kompas Eksternal	SDA
18	Modul Gyro	SDA
19	Modul Gyro	SCL

3. HASIL DAN PEMBAHASAN

3.1. Implementasi

Pada tahap implementasi, penulis akan menerjemahkan aturan-aturan yang ada pada Protokol Dynamixel 2.0 ke dalam bentuk source code dengan bahasa pemrograman C. Adapun tools yang digunakan yaitu Arduino IDE. program yang sudah dibuat tersebut kemudian disebut firmware yang bertugas mengontrol perangkat keras. Firmware yang

telah dibuat akan ditanam pada board Krakatau Sub Controller. Adapun board Krakatau Sub Controller dapat dilihat pada Gambar 4.



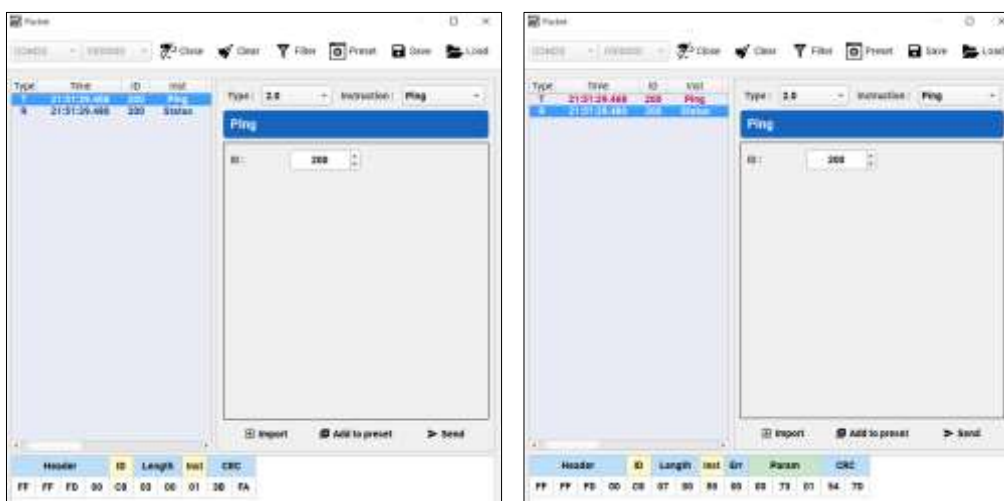
Gambar 4 Board Krakatau Sub Controller

3.2. Pengujian Fungsional

Pada pengujian fungsional, penulis menguji sistem dengan mengirim nilai biner yang ada pada tabel Paket Instruksi. Skema pengujian ini dilakukan dengan mengirim nilai paket instruksi protokol Dynamixel 2.0. Pengujian fungsional akan dianggap berhasil apabila KSC dapat mengirim nilai paket status sesuai dengan paket instruksi.

3.2.1. Pengujian Ping (0x01)

Instruksi Ping merupakan Instruksi yang bertujuan untuk memeriksa keberadaan perangkat dan informasi dasar. Tanpa memperhatikan Status Return Level dari perangkat, Paket Status selalu dikirim ke Instruksi Ping. Adapun paket Instruksi yang dikirim pada pengujian Ping dapat dilihat pada Gambar 5a. Sedangkan paket status dari instruksi Ping yang dikirim oleh KSC V2 dapat dilihat pada Gambar 5b.



(a)

(b)

Gambar 5 (a) Paket Instruksi Ping (b) Paket Status Ping

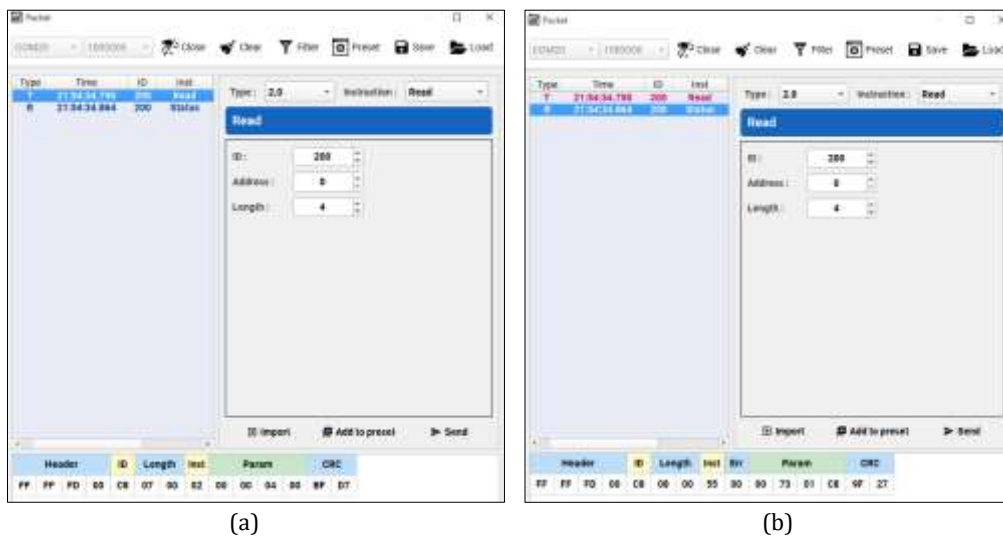
Pada gambar 5a, tampak bahwa nilai yang dikirim ke KSC V2 yaitu sebanyak 10 data dengan rincian 4 data pertama sebagai Header, 1 data sebagai ID perangkat, 2 data sebagai Panjang Data, 1 data Instuksi, dan 2 data Checksum. Pada gambar 5b tampak bahwa Paket Status yang dikirim oleh KSC V2 yaitu sebanyak 14 data. Hal pertama yang perlu diperhatikan bahwa nilai Header yang digunakan bernilai tetap, yaitu 0xFF, 0xFF, 0xFD, dan 0x00. Field Instruksi bernilai 0x55 (Return) yang menandakan bahwa paket tersebut merupakan paket balasan dari perangkat. Adapun penjelasan dari nilai parameter pada Paket Status dapat dilihat pada Tabel 3.

Tabel 3 Parameter pada Pengujian Ping

Parameter Ke	Deskripsi	Nilai
P1	Nomor Model LSB	0x00
P2	Nomor Model MSB	0x73
P3	Versi dari Firmware	0x01

3.2.2. Pengujian Read (0x02)

Instruksi Read merupakan instruksi yang digunakan untuk membaca nilai dari Control Table pada address tertentu. Metode yang digunakan pada pengujian Read yaitu expressing negative number data. Paket Instuksi yang dikirim pada pengujian Read dapat dilihat pada Gambar 6a. Paket Status dari instuksi Read yang dikirim oleh KSC V2 tampak pada Gambar 6b.



Gambar 6 (a) Paket Instuksi Read (b) Paket Status Read

Pada pengujian Read, nilai yang ingin diketahui berada pada alamat 0x00 sampai 0x04. Berdasarkan gambar di atas, penjelasan parameter dapat dilihat pada Tabel 4.

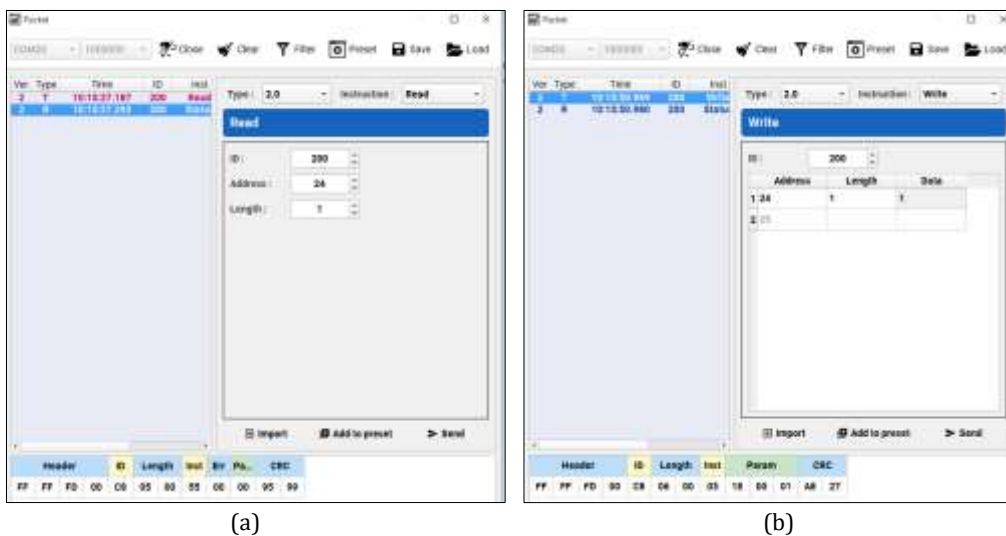
Tabel 4 Parameter pada Paket Status Read

Parameter Ke	Deskripsi	Nilai
P1	Address ke 0	0x00
P2	Address ke 1	0x73

Parameter Ke	Deskripsi	Nilai
P3	Address ke 2	0x01
P4	Address ke 3	0xC8

3.2.3. Pengujian Write (0x03)

Instuksi Write merupakan instuksi yang digunakan untuk menulis nilai pada Control Table. Pada Instruksi Write dieksekusi segera ketika Paket Instruksi diterima. Adapun metode yang digunakan pada pengujian Write yaitu expressing negative number data. Pada pengujian write, address yang akan diubah nilainya yaitu address 0x24 (Dynamixel Power). Adapun nilai sebelum di lakukan instuksi write dapat dilihat pada Gambar 7a. Sedangkan instuksi write dapat dilihat pada Gambar 7b.



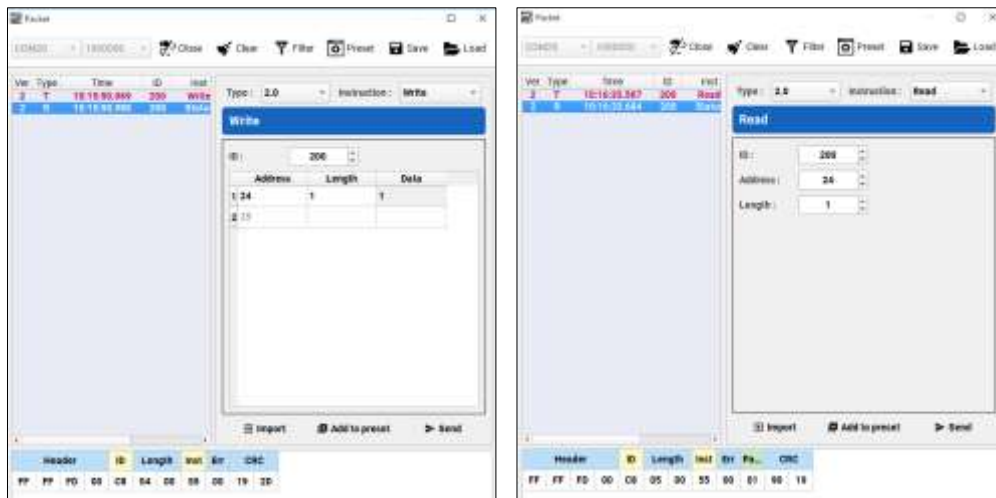
Gambar 7 (a) Address 24 Sebelum di Write (b) Instruksi Write

Berdasarkan gambar 7a diatas, dapat dilihat pada field Parameter nilai yang dibaca yaitu 0x00. Nilai ini merupakan nilai yang dimiliki oleh address 0x24 sebelum dilakukan instuksi write. Tahap selanjutnya yaitu akan dilakukan instuksi write dengan mengirim nilai 0x01 pada address 0x24. Adapun penjelesaian dari parameter pada gambar 4.11 dapat dilihat pada Tabel 5.

Tabel 5 Parameter Instruksi Write

Parameter Ke	Deskripsi	Nilai
P1	Address yang dituju (LOW)	0x18 (HEX) 0x24 (DEC)
P2	Address yang dituju (HIGH)	0x00 (HEX) 0x00 (DEC)
P3	Value	0x01 (HEX) 0x01 (DEC)

Adapun status paket yang dikirim oleh KSC V2 pada pengiriman instuksi write dapat dilihat pada Gambar 8a. Sedangkan instuksi read pada address 0x24 (DEC) setelah dilakukan instuksi write dapat dilihat pada Gambar 8b.



(a)

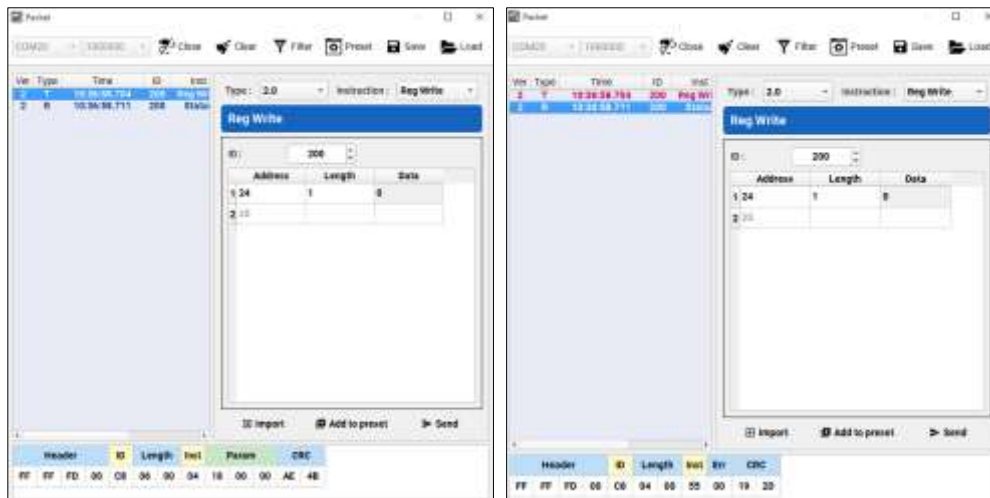
(b)

Gambar 8 (a) Status Paket Instruksi Write (b) Instuksi Read Setelah Dilakukan Instuksi Write

Pada status paket instuksi write, nilai yang perlu diperhatikan yaitu pada field error. Dapat dilihat pada gambar 8a field error, nilai yang terbaca yaitu 0x00 yang berarti tidak terjadi kesalahan pada pengiriman. Tahap selanjutnya yaitu membaca ulang address yang sudah dilakukan instruksi write yaitu address 0x24 (DEC). Proses ini dilakukan dengan tujuan untuk mengetahui apakah nilai yang sudah di write benar-benar tersimpan di address 0x024 (DEC) atau tidak. Berdasarkan Gambar 8b, dapat dilihat pada field Parameter nilai yang terbaca yaitu 0x01. Nilai ini didapat setelah dilakukan instruksi write. Oleh karena itu, dapat disimpulkan bahwa pengujian instruksi write berhasil.

3.2.4. Pengujian Reg Write (0x04)

Instuksi Reg Write merupakan instruksi yang mirip dengan Instruksi Write, tetapi memiliki karakteristik sinkronisasi yang ditingkatkan. Dengan menggunakan Instuksi Reg Write dan Instuksi Action, penulis dapat mengoperasikan beberapa perangkat secara bersamaan. Instruksi Reg Write mendaftarkan Paket Instruksi ke status siaga, dan mengatur Control Table Registered Instruction ke '1'. Ketika Instruksi Action diterima, maka Paket yang telah terdaftar akan dieksekusi dan mengatur Control Table Registered Instruction ke '0'. Adapun Paket Instruksi yang dikirim pada pengujian Reg Write dapat dilihat pada Gambar 9a. Sedangkan paket status pada pengujian Reg Write dapat dilihat pada Gambar 9b.



(a)

(b)

Gambar 9 (a) Paket Instuksi Reg Write (b) Paket Status Instuksi Reg Write

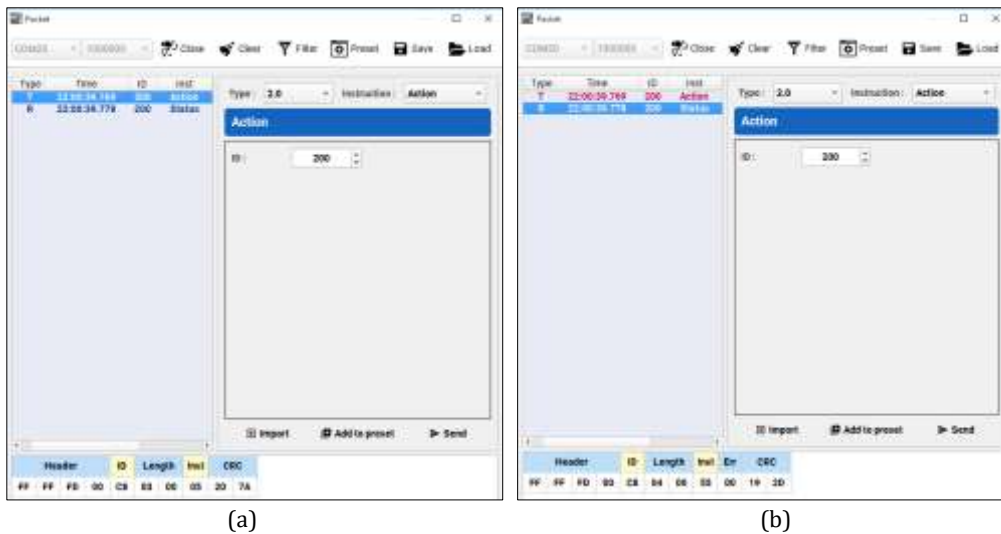
Berdasarkan gambar 9a di atas, terdapat beberapa parameter yang digunakan yaitu P1, P2, dan P3. Adapun penjelasan dari masing-masing parameter tampak pada Tabel 6.

Tabel 6 Parameter Reg Write

Parameter	Deksripsi
P1	Byte low dari alamat awal
P2	Byte high dari alamat awal
P2+1	Byte pertama

3.2.5. Pengujian Action (0x05)

Instuksi Action merupakan instuksi yang digunakan untuk mengeksekusi paket yang telah didaftarkan menggunakan Instruksi Reg Write. Ketika mengontrol beberapa perangkat menggunakan Instruksi Write, akan ada perbedaan waktu eksekusi antara perangkat pertama yang menerima Paket dan perangkat terakhir yang menerima Paket. Oleh karena itu dibutuhkan Instuksi Reg Write dan Instuksi Action agar eksekusi paket dilakukan dalam waktu yang bersamaan. Adapun Paket Instuksi yang dikirim pada pengujian Action dapat dilihat pada Gambar 10a. Sedangkan nilai dari Paket Status yang dikirim oleh KSC V2 dapat dilihat pada Gambar 10b.



Gambar 10 (a) Paket Instuksi Action (b) Paket Status Action

3.2.6. Pengujian Factory Reset (0x06)

Instruksi Factory Reset merupakan instruksi yang bertujuan untuk mengatur ulang Control Table ke pengaturan default pabrik awal. Saat Instruksi Factory Reset (0x06) dilakukan, perangkat di-boot ulang dan LED berkedip empat kali berturut-turut. Jika ID Paket adalah ID broadcast (0xFE) dan parameter opsi adalah Reset All (0xFF), maka Instruksi Factory Reset (0x06) tidak akan diaktifkan. Fitur ini diterapkan dari MX(2.0) FW42, X-series FW42 atau lebih tinggi. Adapun nilai dari instruksi Factory Reset (0x06) dapat dilihat pada Gambar 11a. Sedangkan paket status pada pengujian Factory Reset dapat dilihat pada Gambar 11b.



Gambar 11 (a) Instruksi Factory Reset (b)

3.2.7. Pengujian Reboot (0x08)

Instruksi Reboot merupakan instruksi yang digunakan untuk me-reboot perangkat. Terdapat beberapa jenis instruksi yang hanya dikhususkan untuk perangkat MX-64 seperti Factory Reset, Reboot dan Clear. Oleh karena itu, ketika KSC V2 mendapat paket instruksi reboot, KSC V2 hanya mengirim paket status tanpa melakukan aksi booting ulang. Paket

status dikirim balik oleh KSC V2 dengan tujuan memberikan informasi bahwa KSC V2 juga mendengar perintah reboot. Adapun Paket Instuksi yang dikirim pada pengujian Reboot dapat dilihat pada Gambar 12a. Sedangkan Paket Status yang dikirim oleh KSC V2 pada pengujian Reboot dapat dilihat pada Gambar 12b.



Gambar 12 (a) Paket Instuksi Reboot (b) Paket Status Reboot

3.3. Pengujian Non Fungsional

Pada jenis pengujian non fungsional, aspek yang akan diuji yaitu latensi waktu. Pengujian latensi waktu akan dilakukan pada setiap pengiriman Paket Instuksi. Menurut [13] latensi adalah waktu yang dibutuhkan data untuk menempuh jarak dari asal sampai ke tujuan lalu kembali ke sumber. Tujuan dari pengujian ini yaitu mengetahui waktu yang dibutuhkan sistem untuk mengeksekusi paket yang diterima dan mengirim paket status. Adapun pengujian latensi waktu pada masing-masing pengiriman tampak pada Tabel 7.

Tabel 7 Hasil Pengujian Latensi

Nilai Instruksi	Deskripsi Instruksi	Latensi
0x01	Ping	12 μ S
0x02	Read	114 μ S
0x03	Write	11 μ S
0x04	Reg Write	7 μ S
0x05	Action	9 μ S
0x06	Factory Reset	14 μ S
0x08	Reboot	16 μ S
0x10	Clear	3 μ S
0x82	Sync Read	8 μ S
0x92	Bulk Read	9 μ S

3.4. Analisis Hasil

Berdasarkan implementasi sistem dan pengujian yang telah dilakukan, maka didapat analisa hasil yaitu:

1. Rangkaian half-duplex yang telah dibuat pada Krakatau Sub Controller dapat berjalan dengan baik dibuktikan dengan terjadinya komunikasi antara Main-Controller dengan KSC V2
2. KSC V2 dapat melaporkan kondisi dengan mengirim paket status dan memiliki nilai error 0x00 yang berarti tidak terdapat error selama proses penerimaan instruksi dan pengiriman balasan
3. Pada pengujian fungsional, seluruh aspek fungsional berhasil diuji
4. Pada pengujian non-fungsional, terdapat perbedaan latensi pengiriman paket. Hal ini dapat dipengaruhi oleh banyaknya data yang dilaporkan oleh KSC V2. Semakin banyak nilai pada parameter, maka waktu yang dibutuhkan paket untuk kembali ke asal juga semakin lama

4. KESIMPULAN DAN SARAN

4.1. Kesimpulan

Berdasarkan hasil pengujian dan analisa yang telah dilakukan oleh penulis, maka dapat ditarik beberapa kesimpulan dari penelitian ini, yaitu:

1. Krakatau Sub Controller Versi 2.0 berhasil dibuat dengan menerapkan rangkaian half-duplex dan protokol komunikasi Dynamixel 2.0.
2. Penulis menggunakan FTDI tipe FT232RL sebagai peubah USB menjadi UART dengan bandwith maksimum yaitu 3 Mbps.
3. Waktu yang dibutuhkan MX-64 untuk merespon perintah yaitu PING selama 1 μ S, READ selama 1 μ S, WRITE selama 1 μ S, ACTION selama 1 μ S, CLEAR selama 1 μ S, REBOOT selama 2 μ S, REG WRITE selama 1 μ S, BULK READ selama 9 μ S, dan SYNC READ selama 8 μ S.

4.2. Saran

Adapun beberapa saran yang dapat digunakan sebagai bahan pertimbangan bagi peneliti selanjutnya yaitu:

1. Menggunakan jenis FTDI lainnya yang memiliki bandwith lebih dari 3 Mbps sehingga servo dengan bandwith yang lebih besar dari 3Mbps dapat dikendalikan
2. Menambahkan rangkaian pengamanan arus, penyetabil tegangan sehingga meminimalisir terjadinya corrupt data.

DAFTAR PUSTAKA

- [1] Wijaya, M.A., Pratomo, A.H., Widodo, N.S., Junaedi, G.S., 2016. Kendali Darwin-Op Pada Robot Humanoid Upn "Veteran" Yogyakarta. Prosiding: Seminar Nasional Informatika. 1(2).
- [2] Setiawan, D., Rosandi, I.S., Putra, M.P.K. And Darmawan, S., 2018. Deteksi Bola Multipola Pada Robot Krakatau Fc. 5th Indonesian Symposium On Robotic Systems And Control.

- [3] Sun, J., Jones, M., Reinauer, S. And Zimmer, V., 2015. Embedded Firmware Solutions: Development Best Practices For The Internet Of Things (P. 224). Springer Nature.
- [4] Robotics Co (2021) Emanuel-Robotis. Available At: Emanuel.Robotis.Com (Accessed: 1 April 2021).
- [5] Mirow, R. And Bestmann, M.S.M., 2020. Embedded Debug Interface For Robots (Doctoral Dissertation, Universität Hamburg).
- [6] Bestmann, M., Güldenstern, J. And Zhang, J., 2019, July. High-Frequency Multi-Bus Servo And Sensor Communication Using The Dynamixel Protocol. In Robot World Cup (Pp. 16-29). Springer, Cham.
- [7] Arámbula, F., Arce, H.R., Gómez, Á.D., Hernández, E., Hunter, M.E., Ibarra, J.M., Malo, A.J., Lara, F.J., Lavín, J.E., Llarena, A. And López, I., 2011. Dotmex (. Mx) Humanoid Kid-Size Team Description Paper.
- [8] Yusuf, H. L. Et Al. (2018) 'Dago Hoogeschool Dalam Kontes Robot Sepak Bola Indonesia Humanoid 2018', In The 6th Indonesian Symposium On Robotic Systems And Control (Isrsc), Pp. 171–175.
- [9] Ha, I., Tamura, Y., Asama, H., Han, J. And Hong, D.W., 2011, September. Development Of Open Humanoid Platform Darwin-Op. In Sice Annual Conference 2011 (Pp. 2178-2181). Ieee.
- [10] Fabre, R., Rouxel, Q., Passault, G., N'guyen, S. And Ly, O., 2016, June. Dynaban, An Open-Source Alternative Firmware For Dynamixel Servo-Motors. In Robot World Cup (Pp. 169-177). Springer, Cham.
- [11] Putra, G.N., Wahab, F.A., Asih, D.S., Surya, M. And Nugraha, A.K., 2018. Rancang Bangun Sistem Tertanam Pada Humanoid Robot Soccer Tim Gladiatos Ui. 5th Indonesian Symposium On Robotic Systems And Control
- [12] Susanto, A., Chayadi, J., Danieal, Dharmawan , A.B., 2018. Rancang Bangun Robot Devata New Generation 2017. 5th Indonesian Symposium On Robotic Systems And Control
- [13] Darmawan, D. And Imanto, T., 2017. Analisa Link Balancing Dan Failover 2 Provider Menggunakan Border Gateway Protocol (Bgp) Pada Router Cisco 7606s. Jurnal Nasional Teknologi Dan Sistem Informasi, 3(3), Pp.326-333.