International Conference on Engineering, Advance Science and Industrial Application (ICETESIA) 2018
September 6-7 2018, Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia

Designing Interface of Rhino Robot’S Operator Control Unit using Development Board LattePanda

Livian Tjandra[1], Achmad Affandi[2], Imam Arifin[1], and M. Fajar Adityo[3]

Abstract*—Rhino Robot work based on the control given by the user through a control device named Operator Control Unit (OCU). Currently, device that has been developed are form laptop toughbook with gamepad as controller. This design were inconvenience user, because there are batch of buttons which are not used to operate the robot. Moreover, using combination keys in gamepad to choose menu operation are required. Therefore, further development for the control device of Rhino Robot is do. The design were used is in OCU consist of an panel input that contains a several buttons and joystick which have a specific function to operate Rhino Robot. The buttons will be placed according to its function to the robot control process, using visualization of Rhino Robot to help user understanding the function of each buttons. This study will discuss the process to making interface of each buttons. In addition, it also discusses the process how to make an interface of LED which used to buttons indicator. The compilation process utilizes the existing LattePanda’s Arduino co-processor using Visual Studio to create a program. After testing of the several buttons using program that have been create, it is known that the program have been able to perform each function of buttons and LED in input panel.

***Keywords*—Rhino robot, OCU, Interface, LattePanda.**

I. INTRODUCTION

Rhino Robot is an EOD (Explosive Ordnance Disposal) Robot were developed by PT Bhimasena Research and Development with the aim to replace human in handling a dangerous objects. This intended to minimize the possibility of the human get an injury. In operation, Rhino Robot is controlled remotely by user through a control device named Operator Control Unit (OCU).

The device that has been developed are form of a toughbook laptop to display the GUI with gamepad as controler. Forms of devices like this are considered less efficient and tend to be complicated. Because on the laptop there are several buttons that are not used. In addition, user are required to know how to used the gamepad’s key combinations to select the menu. In practice, however, most users want more efficient and easy-to-operate devices.

Advanced development for the Rhino Robot’s control device has a design there is an input panel, which contains a number of buttons equipped with an indicator LED. Where each button has a specific function. The buttons are placed according to their controled function on the robot, with help of the visualization by the image of the Rhino Robot. To be able to perform the functions of these buttons, the interface process is needed. The method used to perform the interface function of the buttons and indicator LED utilizes a development board named LattePanda which has been integrated by the Arduino co-processor [1]. The interface created using C# programming language that runs over the .NET framework found in Visual Studio.

II. BACKGROUND

EOD robot in general has a physical shape as in Figure 1. This robot has an aims to replace humans by handling some dangerous objects [2]. This intended to minimize the possibility of the human get an injury. In practice, these robots still require a constant attention and control from the user. This can be solved by using a device called OCU [3] as shown in Figure 2.

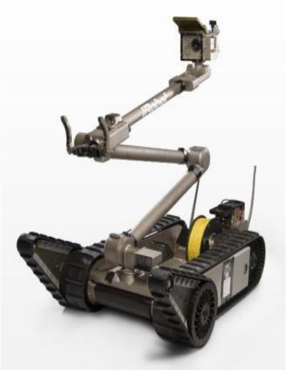


Figure 1. EOD Robot.

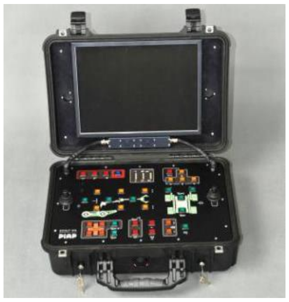


Figure 2. Operator Control Unit.

[1]Livian Tjandra and Imam Arifin are with Department of Automation Electrical Engineering, Institut Teknologi Sepuluh Nopember, Surabaya, 60111, Indonesia. E-mail: livlivian98@gmail.com.

[2]Achmad Affandi is with Department of Electrical Engineering, Institut Teknologi Sepuluh Nopember, Surabaya, 60111, Indonesia.

[3]M. Fajar Adityo is with Staff of PT Bhimasena Research and Development.

OCU is usually equipped an input panel consisting some of buttons and joysticks to operate the robot [4]. The design of the external device are using the working principle of the user interface [5]. The devices that used in the interface process are LattePanda and several modules, such as the push button board, LED board, and ATMega16 board. The preparation of the interface is done by using C# programming that exist in Visual Studio.

A. LattePanda

The first development module that can run the full version of Windows 10 called LattePanda. This module operates turbocharged through an Intel Quad Core 1.8 GHz processor, with 32-64 GB of RAM and 2-4 GB RAM. Supported by Arduino co-processor titled plug and play and GPIO (general-purpose input / output) pin, it is able to access standard sensor and actuators operating at 5V voltage [1].

To accessing Arduino's GPIO, it takes a LattePanda.Firmata library which is an open-source library from Firmata. This library is provided by the LattePanda development board in accordance with Windows applications that has been developed in Visual Studio [6].

B. Board Ocu

The several modules that used in designing an interface of Rhino Robot's OCU there are push button board, LED board, adn ATMega16 board.

In its application, the push button board is intended to handle the use of microcontroller pins in small quantities while the buttons are used in large quantities. The work done by this board in accordance with the main component of the compilers, the multiplexer CD4067B. To connect the board with LattePanda and each buttons, several connectors are used.

LED board is used to handle LED inputs that require using a few microcontroller pins. This board works by utilizing the function of transistor switches that has been provided on the board, which makes 1 input data received can be used for a numbers of LED.

The function of ATMega16 is to run USART communication between LattePanda and LED. This communication aims to send information from the state of the button on the input panel that has been processed through the indicator LED.

C. C# Programming

C-sharp (C#) is a new programming language created by Microsoft along with the creation of .NET framework. Chief Architect in making C# is Anders Hejlsberg who previously make Borland Delphi and Turbo Pascal. C# is based on C and C++ languages, so aspects such as statements, expressions, operators, etc, with various improvements that make the language simpler.

C# is organized by namespace, that make it can be used as an internal organizational system and organized externally. Which makes it accessible publicly using "using" keyword. According to ECMA, classes are data structures that contain data (constant and field), functions (methods, properties, events, indexers, operators, constructors, destructor, and static constructors), as well as classes in classes. In C#, all programs are built on the class. One special iteration in C# that doesn't come from C is called "foreach". This form is an adaptation of "for each" that is in Visual Basic. The foreach statement is used to trace a collection.

III. Implementation

The interface on the Rhino Robot's OCU is designed by utilizing the features already provided by the Arduino co-processor on LattePanda [1]. The co-processor is used to perform the functions of the constituent boards, including push button boards, LED boards, and ATMega16 boards. The boards are processed through programs created in Visual Studio, with the illustration as shown in Figure 3.

The program is constructed by using C# programming in Visual Studio like a data flow diagram as shown in Figure 4. Which the device used as control is gamepad and arduino (input panel), the information transmitted is then processed on control handler, control center, and input system also led controller that serves to send information that can make LED indikator on.

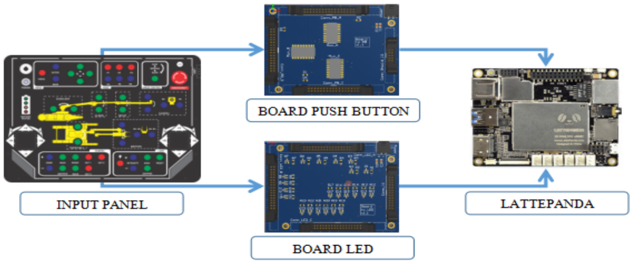


Figure 3. Illustration of Interface Design of Rhino Robot's OCU.

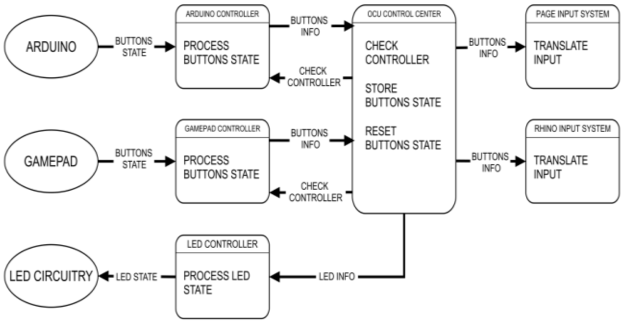


Figure 4. Data Flow Diagram of Rhino Robot'sOCU Interface Program.

A. Control Handler

In section called control handler, the program performs a function to check the connections of each control device. After device connectivity is known, a declaration of the button state is performed. The declaration is done by creating a method to update from the state of the buttons and return the buttons value to default state.

The data in the control handler is processed by different program files, according to the connected device. The information sent by the gamepad is processed on GamepadController.cs and information from Arduino (input panel) is processed in ArduinoController.cs.

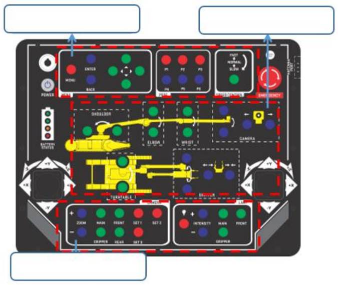


Figure 5. Input Panel.

Because of the input panel processed 45 buttons with a few Arduino's pin usage, a special handling is required. Therefore, a push button board with MUX as its main component are used. Which each MUX on the board serves to process 16 input data. Grouping of buttons is done based on its location on the input panel, as in Figure 5.

The first MUX is used to process input data from a group of buttons located at the top of the panel. The buttons has a function to access the menu buttons.

The middle part of the panel is a group for the buttons to be accessed using the second MUX. The buttons in this group are groups of buttons used to select the operation menu of the robot, ranging from the movement of the robot arm to the movement of the camera and the handheld used.

The last MUX on the board is used to perform group button processing located on the bottom panel. The buttons incorporated in this group are the accessories buttons of the robot, including buttons to set the camera and lights on the robot.

Beside grouping the buttons on the panel to note, ArduinoController.cs also requires an additional function to know the state of the LattePanda's port connection. But in the LattePanda.Firmata namespace, aren't function that can run the command. Therefore, an additional function in the Arduino.cs file are created. The function is named IsConnecter ().

B. Control Center

Information that has been processed in control handler will be contained in the control center. As long as the information is in the control center, the program will check which device is active (connected). Afterwards, the program will store the information sent by the control handler and change the status information of buttons on default state.

When a control device check is performed, a method is needed to deal with the possibility of connected enhancements. The method contains a command to know the state of the device. To access the methode, a new class called AuxiliaryControl is required. The class contains declarations for the circumstances that may occur in additional csniontrol devices. Starting from the function to determine the state of the device is active or not up to the mode of the device. This class is placed in an additional program file containing the enumeration groups used.

C. Input System

In the input system, the information sent by the control center is translated to be able to perform the expected function. This information translation process is performed on two different program files, namely PageInputSystem.cs and RhinoInputSystem.cs.

PageInputSystem.cs process to translate information related to the operation page (GUI). The buttons are used in this process is accordance on Figure 6.

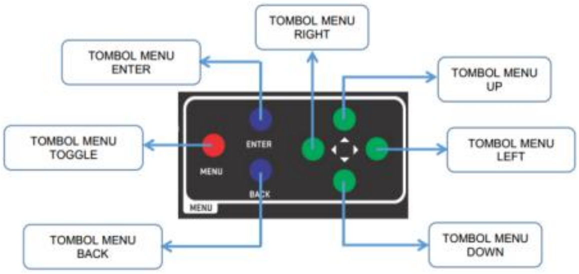


Figure 6. Empty groups accessed by PageInputSystem.cs

RhinoInputSystem.cs file serves to translate the function buttons which to operate robot. The buttons accessed by this program are buttons that are not accessed in the PageInputSystem.cs file. In this process there are some special handling for accessing certain functions. This handling is done due to the use of a combination of buttons, including ToggleDrivetrainMode function and ToggleBrake function.

In addition translating information for each buttons, input system also has a process to perform toggle data sent by LEDController.cs. The LED toggle data process is also performed on two different files. Data accessed on PegeInputSystem.cs includes data for emergency, toggle menu, state menu, and camera. RhinoInputSystem.cs file accesses data for emergency, normal, inv. kin, and lamp.

D. LED Controller

The commands that executed in the LEDController.cs program file start from the declaration of the port used, like name, baudrate, and other needed. The next process is checking port connectivity. Then proceed with declarations for data updating methods in accordance with the latest information sent by the control device, as well as methods for returning data to the default state.

In the process of sending data, handling is needed to set the value of each bit, or it can be called by setting the state of each LED. The handling consists of two methods, including to set the bit value being one (1), which means to set LED turn on and return the bit being zero (0) to set the LED turn off.

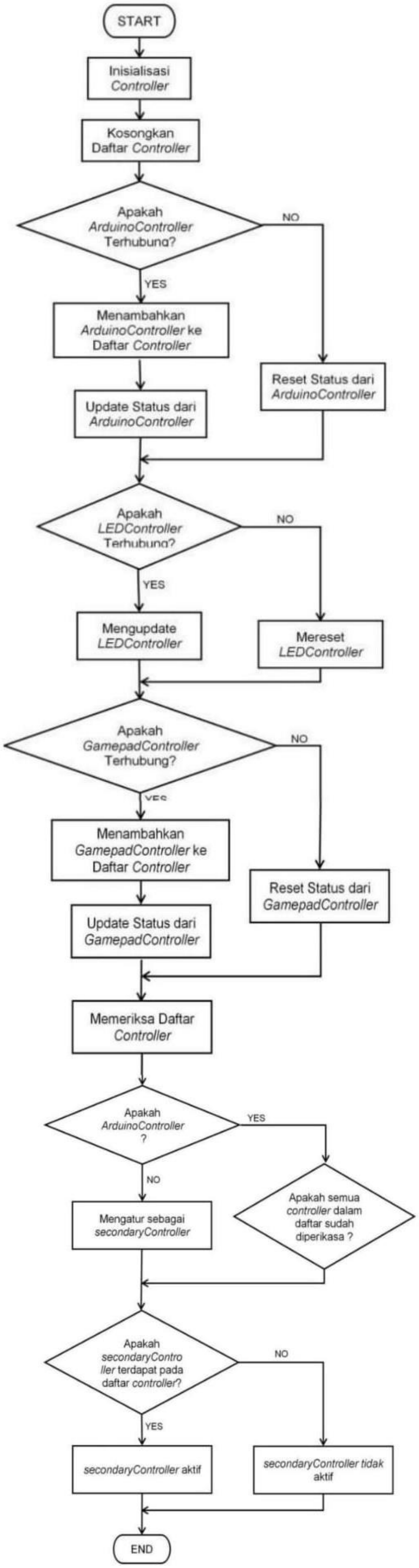

START
Inisialisasi Controller
Kosongkan Daftar Controller
Apakah ArduinoController Terhubung?
NO
YES
Menambahkan ArduinoController ke Daftar Controller
Reset Status dari ArduinoController
Update Status dari ArduinoController
Apakah LEDController Terhubung?
NO
YES
Mengupdate LEDController
Mereset LEDController
Apakah GamepadController Terhubung?
NO
YES
Menambahkan GamepadController ke Daftar Controller
Reset Status dari GamepadController
Update Status dari GamepadController
Memeriksa Daftar Controller
Apakah ArduinoController ?
YES
NO
Mengatur sebagai secondaryController
Apakah semua controller dalam daftar sudah diperikasa ?
Apakah secondaryController terdapat pada daftar controller?
NO
YES
secondaryController aktif
secondaryController tidak aktif
END

Figure 7. Flowchart program interface OCU Rhino Robot.

TABLE 1.
PROTOKOL USART LATTEPANDA WITH ATMEGA

Byte_Index	Contain	Info	Bit_Index
0	Header	0x40 (‘@’)	n/a
1	Data_Size	2	n/a
2	Status	~	7
		Inv.Kin	6
		Normal	5
		Emergency	4
	Menu	~	3
		~	2
		Menu Status	1
		Menu Toggle	0
3	Cam_Select	Rear	7
		Front	6
		Gripper	5
		Main	4
	Lamp_Select	Front	3
		Gripper	2
		Main	1
		Status	0
4	CRC(XOR)	XOR_Status	n/a

The process of setting bit being 1, an OR operation is used between the bearing data value and the declared variable (the variable is then named mask). As for returning the bit being 0, an AND operation is used between the bearing data and the result of the value reversal of the mask. However, the use of a function for value-biting (operator complement bitwise) in Visual Studio causes a wrong reading value (causing the end result of data to be negative). It is therefore necessary to use unchecked functions on the AND process. The commands are accessed using UART communication through a protocol that has been created, the data lists submitted by the protocol in accordance with the list in Table 1.

E. *Flowchart*

Overall, the program from the Rhino Robot’s OCU input panel interface that has been compiled has an access flow according to the flowchart in Figure 7. Where the operating center of the program is in the OCUControlCenter.cs program file which is responsible for checking the connected controller.

IV. EXPERIMENT

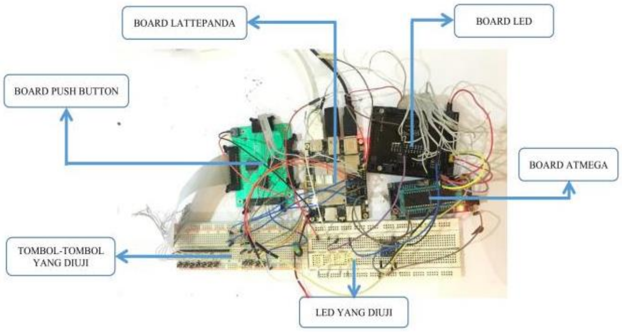
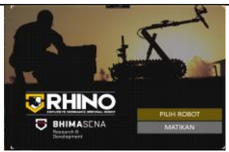
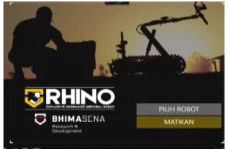
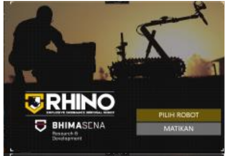
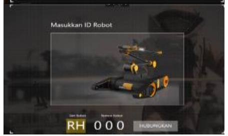
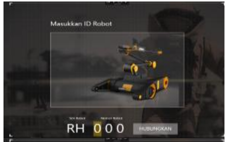
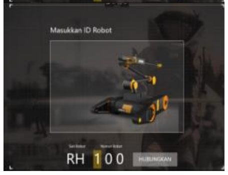
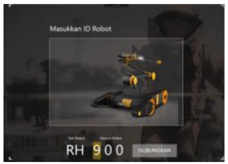
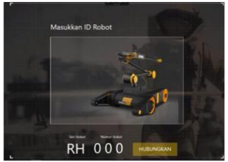
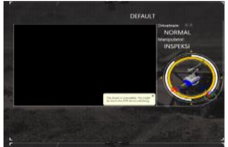
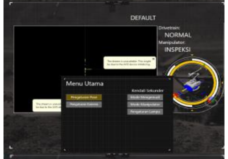
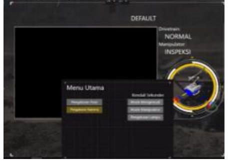
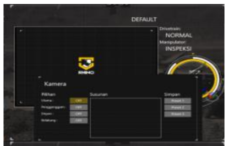
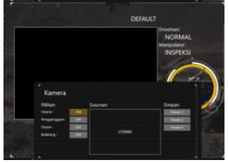
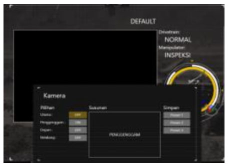
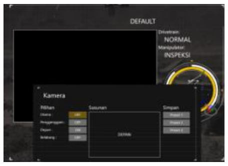
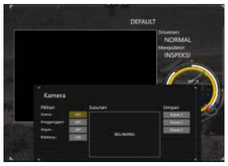


Figure 8. OCU Rhino Robot Interface Testing Circuit.

TABLE 2.
BUTTONS INTERFACE TEST RESULT ON THE INPUT PANEL

No	Experiment	Result	Explanation
1	Running the program		Displays the main page of the GUI, where the cursor is in the default state.
2	Press down button		The cursor moves to the options menu underneath.
3	Press up button		The cursor moves to the options menu above it.
4	Press enter button		The displayed page moves to the next page.
5	Press right button		The cursor moves to the selection menu on the right.
6	Press up button		The numbers displayed in the options menu do the calculations of the smallest number (1).
7	Press down button		The number displayed in the options menu calculates the largest number (9).
8	Press left button		The cursor moves to the selection menu on the left.
9	Press enter button		The displayed page moves to the next page.
10	Press toggle button		A pop-up menu page will appear.
11	Menekan tombol menu *down*		The cursor moves to the options menu underneath.
12	Menekan tombol menu *enter*		The pop-up menu page displayed changes to the next page.
13	Menekan tombol menu *camera main*		The state of the main camera menu turns to ok, and the video shown is the information sent by the main camera.
14	Menekan tombol menu *camera gripper*		The cursor does not change position, but the state of the camera gripper menu turns to ok, and the video to be displayed is information sent by the handheld camera.
15	Menekan tombol menu *camera front*		The state of the camera front menu changed to ok, and the video shown is the information sent by the rear camera.
16	Menekan tombol menu *camera rear*		The state of the camera rear menu turns to ok, and the displayed video is the information sent by the front camera.

The circuit used for testing looks like in Figure 8, where LattePanda is connected to the three OCU input panel compiler boards, namely the push button board, LED board, and ATMega board. The push button board is then connected directly to the buttons used. Likewise with the LED board, directly connected with LEDs that act as indicators.

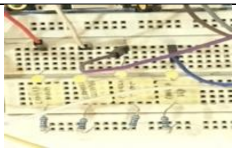
A. Push Button

The test performed to determine the accuracy of the function of the interface program the buttons on the input panel in accordance with Table 2.

B. LED

Testing the interface for the indicator LED that is done is to observe the state of the LED from the camera main menu button, the camera gripper menu, the camera front menu, and the camera rear menu. The process is done by setting the state of some cameras through the menu buttons that have been provided. The results obtained from the test can be seen in Table 3.

TABLE 3.
RESULTS OF LED INDICATOR INTERFACE TEST

No	Langkah	Hasil yang didapatkan	Keterangan
1	There are no button is pressed		No indicator LED lights up
2	Press camera rear button		The state of the front camera on the display turns 'on' and the indicator LED from the camera rear menu button lights up
3	Press camera front button		aims to prove if the button can be used more than 1 at a time

V. CONCLUSION

The results of the tests indicate that the compiled system has been able to perform the desired interface function. It is known from the test results that indicate when the button is pressed, the interface function played is then executed. In addition, the LED is used as an indicator of several buttons have been able to run according to function, which when the button is pressed, then the LED that acts as an indicator will light up.

REFERENCES

[1] Lattepanda.com, “Peripheral Device.” [Online]. Available: http://docs.lattepanda.com/content/getStarted/peripheral/.

[2] F. Luqiao and et al, “A Single-Hand and Binocular Visual System for EOD Robot,” in *Automation and Logistics, 2007 IEEE International Conference on*, 2007.

[3] N. Checka and et al, “Handheld Operator Control Unit,” in *Human-Robot Interaction (HRI), 2012 7th ACM/IEEE International Conference*, 2012.

[4] J. Crossman, R. Marinier, and E. B. Olson, “A Hands-Off, Multi-Robot Display for Communicating Situation Awareness to Operators,” in *Collaboration Technologies and Systems (CTS), 2012 International Conference*, 2012.

[5] P. Candela, “Designing an Operator Control Unit for Cooperative ` Unmanned Systems,” in *Proc. SPIE*, 2017.

[6] Lattepanda.com, “3 Steps to Access Pinouts.” [Online]. Available: http://docs.lattepanda.com/content/hardware/accessPinoutsFromVS/.