2010/0010
PERPUSTAKAAN
POLITEKNIK MANUFAKTUR
BANDUNG
25 MAR 2010

Report on Overseas Non Degree Training Program Indonesia Managing Higher Education for Relevancy and Efficiency (IMHERE)

"APPLICATION of CAN BUS for FW 36 REMOTE CONTROL BASED ON MATLAB/SIMULINK "

Written by
Wahyudi Purnomo

Supervisor
Dipl.-Ing. W. Lalo
Prof. Dr.-Ing. D.Schramm

UNIVERSITÄT DUISBURG ESSEN

UNIVERSITY OF DUISBURG-ESSEN, GERMANY
DECEMBER 2008

CAN BUS APPLICATION for the FW 36 REMOTE CONTROL WITH MATLAB/SIMULINK

A. INTRODUCTION

The objective of this practicum is to design and realize the communication between the "FW 36 Remote Control" and a PC via Controller Area Network (CAN) Bus (in hard real-time environment). The FW 36 Remote Control is used to control the manipulator movement of Truck-mounted concrete pumps ofSchwing Company .The control system is designed in the Matlab/Simulink environment.

With the growing importance of timing accuracy in simulating control systems during design and development, there is an increased need for these simulations to run in a real-time environment. This practicum further more focuses on software tools that support hard real-time environments to run real-time simulations.

In this system, Matlab/Simulink was selected as real-time computer aided control design software for demonstrating the real-time simulation.

B. ACKNOWLEDGMENT

This practicum was sponsored by a grant from Indonesia-Managing Higher Education for Relevance and Efficiency (I-MHERE) program

Base Time : October – Desember 2008

Reference : Mechatronics Lab (Departement of Mechatronics) – University of Duisburg Essen

C. MATLAB/SIMULINK

Software tools such as Matlab/Simulink (and programming tools) are essential in the field of control systems design, modeling, and simulation. where It provides mathematical programming and a graphical modeling platform for designing and simulating systems.

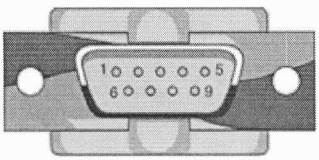
The Real-Time Workshop, a MATLAB/Simulink add-on for automatic code generation, can be used for general Microsoft Windows based real-time applications with the help of the Real-Time Windows Target

To implement the system in hardware, a communication network such as Controller Area Network (CAN) is required for communication among agents. Hence, the real-time system should support the interface with this communication infrastructure

D. PEAK CAN USB DONGLE

Out of various CAN-PC interface devices available today, the USB-CAN device from Peak-Systems was chosen as it is portable for ofther platform besides Windows.
The device is based on a Phillips SJA1000 CAN Controller with 16MHz clock frequency, 82C251 CAN transceiver, and has a 9-PIN connector. Figure 3.1 shows a picture of the PCAN-USB device and its connector.

Pin assignment 9-pole connector male:



Pin Configuration	
1	Not connected / +5V optional
2	CAN-L
3	CAN-GND
4	Not connected
5	Not connected
6	CAN-GND
7	CAN-H
8	Not connected
9	Not connected / +5V optional

*Figure 1: PCAN-USB device with 9-PIN connector

E. SIMULINK S-FUNCTION BLOCK DESIGN of CAN USB

Simulink blocks are developed to interface the PCAN device to receive and send CAN messages. The block shown in Fig. 2 is used for receiving CAN messages. The block has output ports each representatives the data of a receive ID. The number of output ports in the "PCAN RECEIVE " block depends on the number of ID's parameter value in the block parameters.

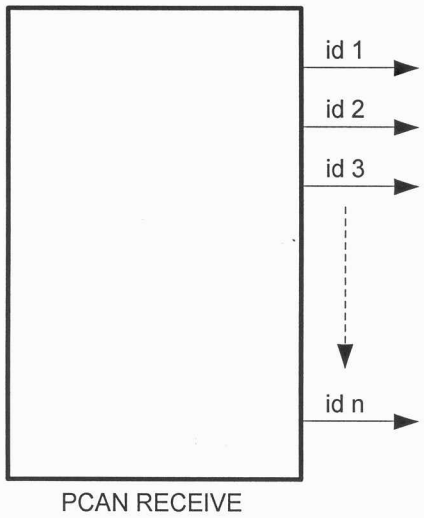


Figure 2. PCAN RECEIVE block.

Fig. 3 shows the "PCAN SEND " block where this block can be used for sending messages to the CAN bus. This block has one input port as input "DATA" port.

Figure 3. PCAN SEND block.

Both blocks have four parameters:

1. IDs number (We can enter one or more as we need (max 10))
2. Baudrate (10Kbps, 20Kbps, 30Kbps, 50Kbps, 100Kbps, 125Kbps, 250Kbps, 500Kbps, and 1Mbps).
3. Sampling Time (in ms)

   Is defined as the number of samples per second
4. Transceive Parameter (Receive or Send)
   - If we choose "*RECEIVE*", the S-Function block becomes a "PCAN RECEIVE" block.
   - If we chose "*SEND*", the S-Function block becomes a "PCAN SEND" block.

Note :

1. *For "CAN RECEIVE", only one module (function block) can be used because the CAN-USB dongle just has 1 internal buffer. If you use the module more than one, may be just one of module that is functioned properly*
2. *For "CAN SEND" can use more 1 modul because the equipment that used have more internal buffer.*

F. PROCEDURE to CREATE S-FUNCTIONS

S-functions provide a powerful mechanism for extending the capabilities of the Simulink environment. An *S-function* is a computer language description of a Simulink block written in MATLAB, C, C++, Ada, or Fortran.

S-functions are compiled as MEX-files using the *mex* commad. As with other MEX-files, S-functions are dynamically linked subroutines that the MATLAB interpreter can automatically load and execute.

Fig. 4 bellow show the different procedures of an S- Function.

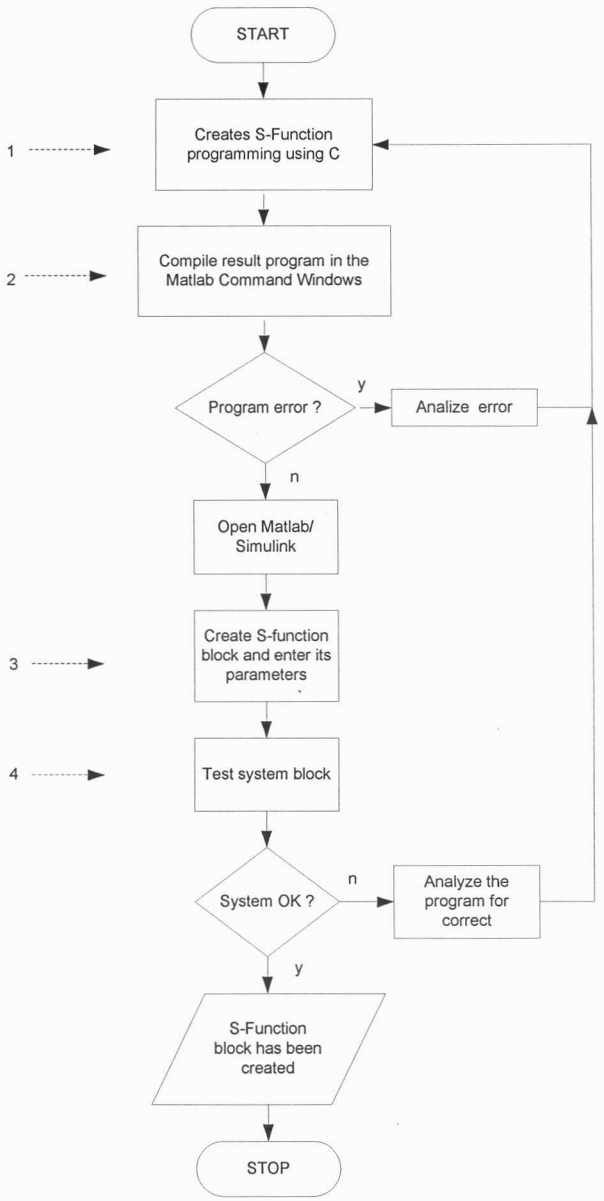


Figure 4. Create S- Function Block procedure

1. Create a program for the S-function block. In this case we use C programming language for developing the CAN Communication (Write program as well as structure s – Function builder).
2. After the program has finished, save program (file name = program name !!). Compile program in Matlab Command Window.

Instruction :

>> mex file_name.c <enter>

3. If no error occurs, you can continus with Simulink.

   Follow the procedure as shown in Fig. 5.

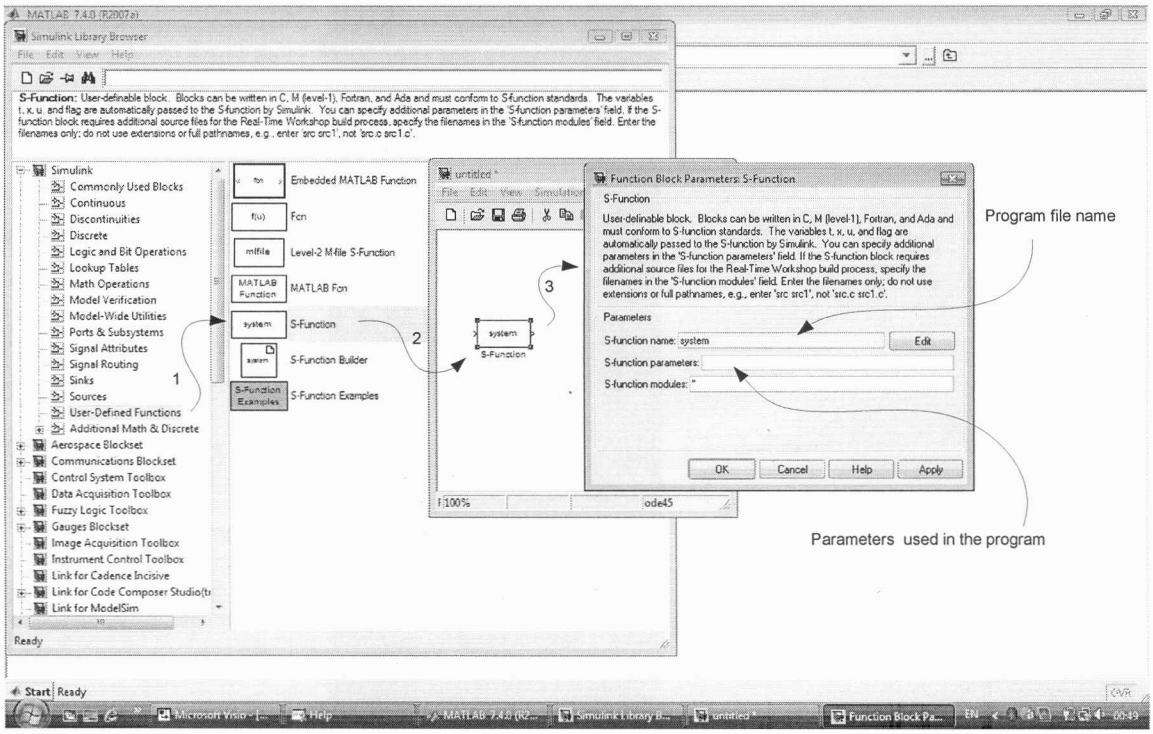


Figure 5. Procedure to realize programs in S-Function blocks.

4. System Block Test.

   Connect the system requirement as shown in Fig. 6.

   Running the system and inspect the output.

   You must reprogram if the system is not working properly until the program results are correct.

Figure 6. Diagram Block for system test

G. REALIZATION OF the CAN USB SIMULINK BLOCK

For this case, we create an S-Function program for CAN USB using C programming language.

Program file name : ***peak_can_usb2a*** . The program is shown as in index a.

After the S-Function program is compiled, the procedure to create S-Function block can be shown in Fig. 7.

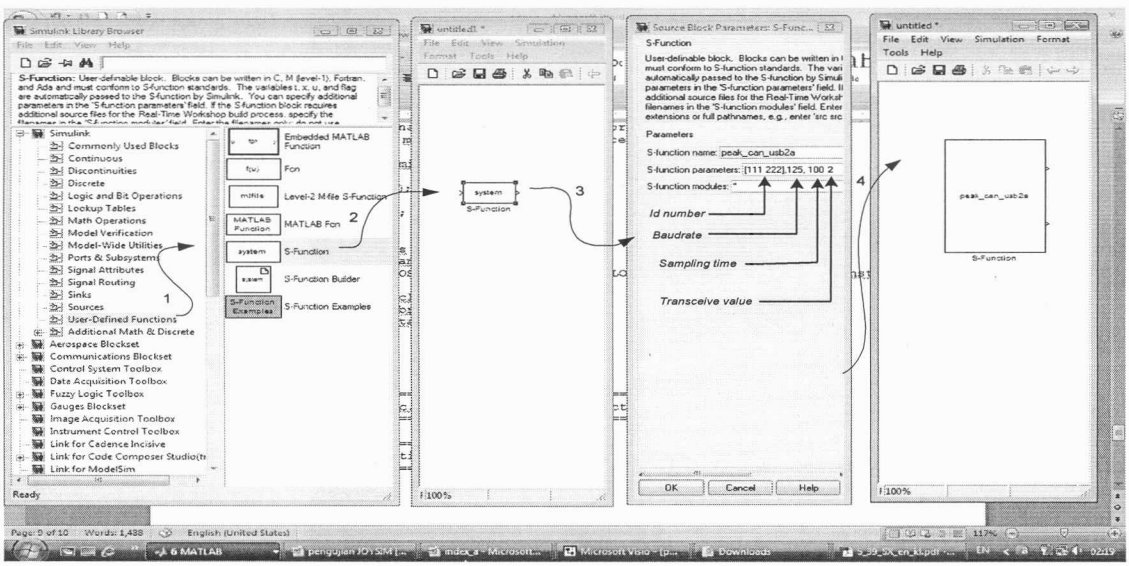


Figure 7. Create S-Function Block

For easier use, the system block can be made a "*source block parameter* " as shown in Fig. 8.

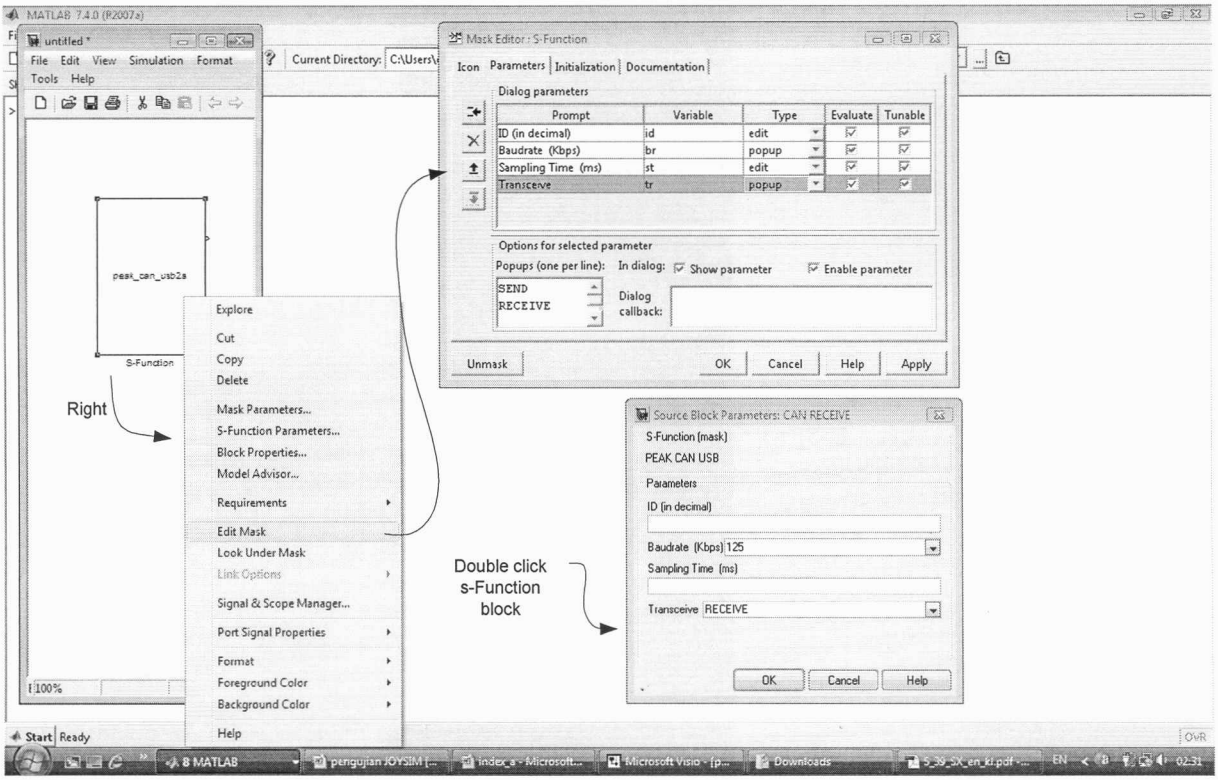
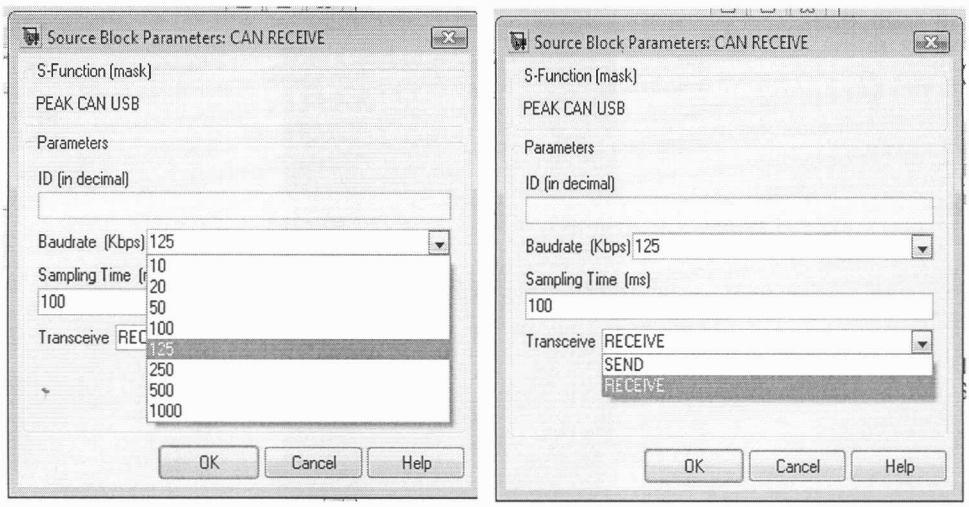


Figure 8. Create Source block parameter of S-Function

G.1 Program Implementation

a. CAN Receive



. Figure 9. Choice baudrate (25Kbps) and transceive value (RECEIVE)

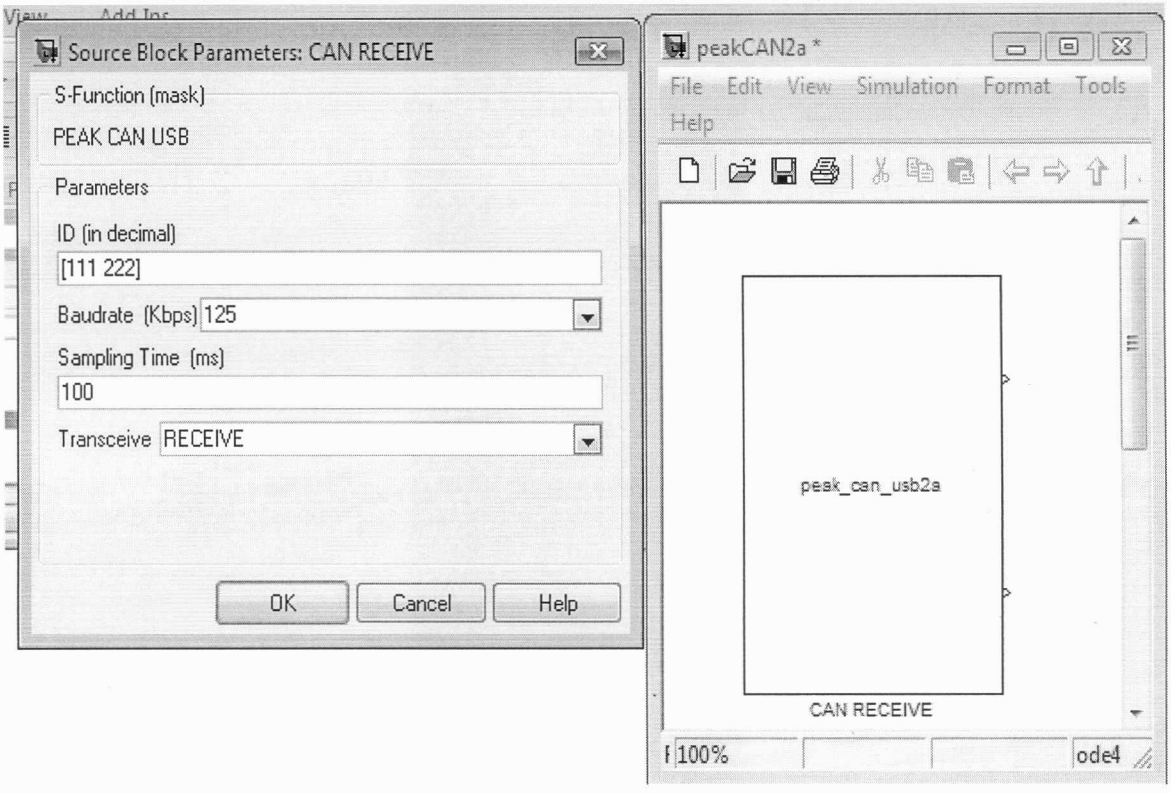


Figure 10. Two ID Outputs (in decimal) and S-Function block result

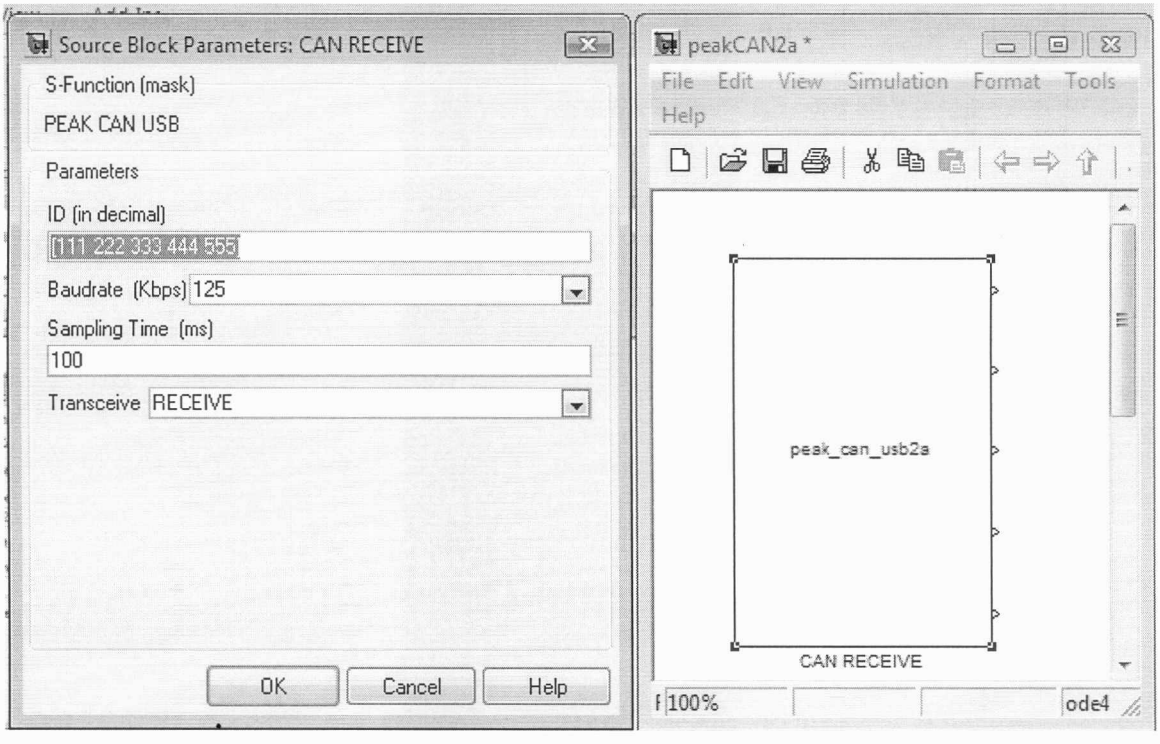


Figure 11. Five ID outputs and S-Function block result

b. CAN Send

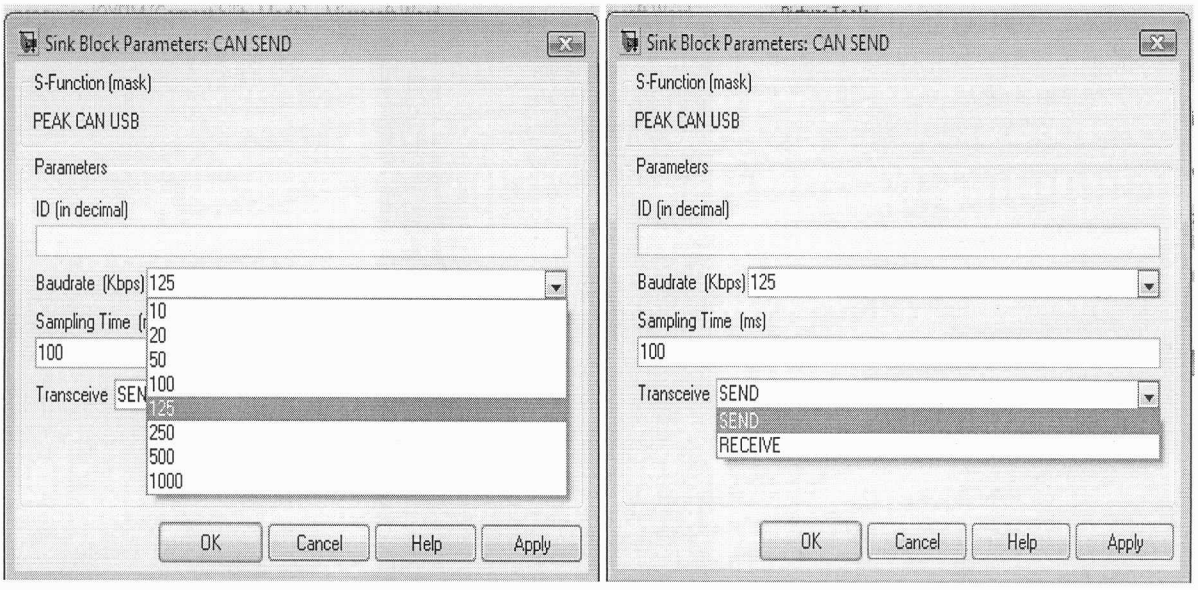


Figure 12. Choose baudrate (25Kbps) and transceive value (SEND)

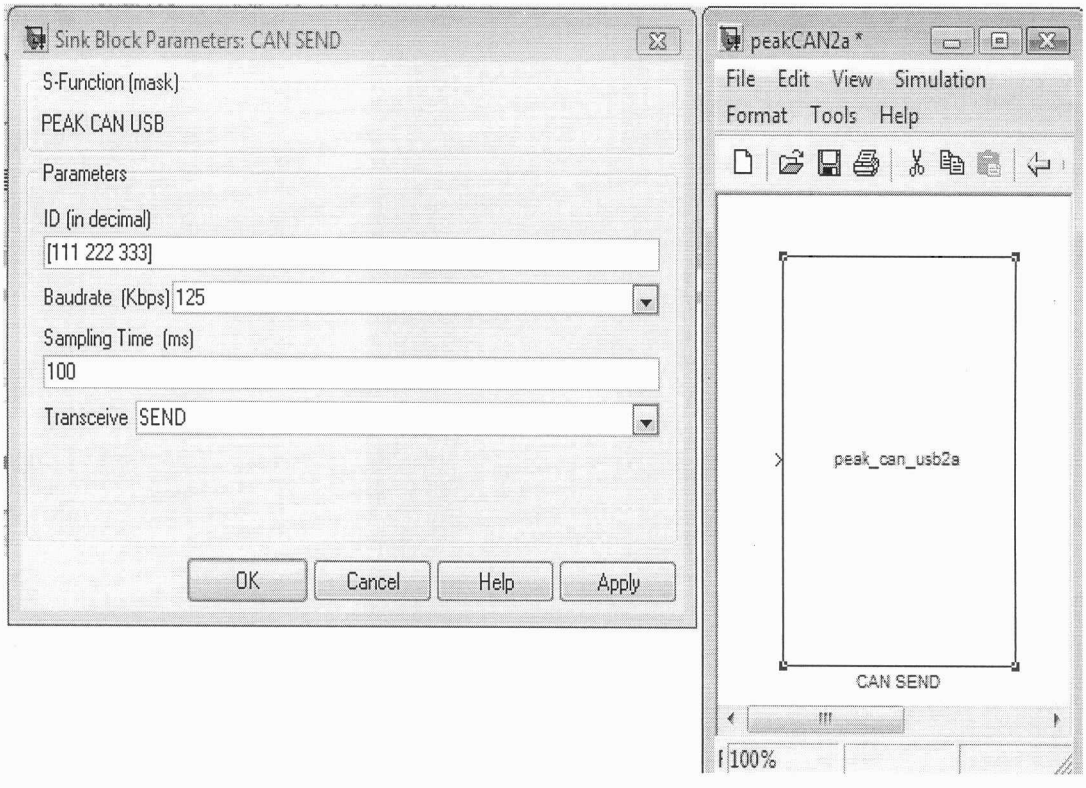


Figure 13. S-Function Block result

G.2 System Function Test

Note :

Before running, you must the instal driver for the CAN-USB dongle on your computer.

a. Block Diagram for Function Test

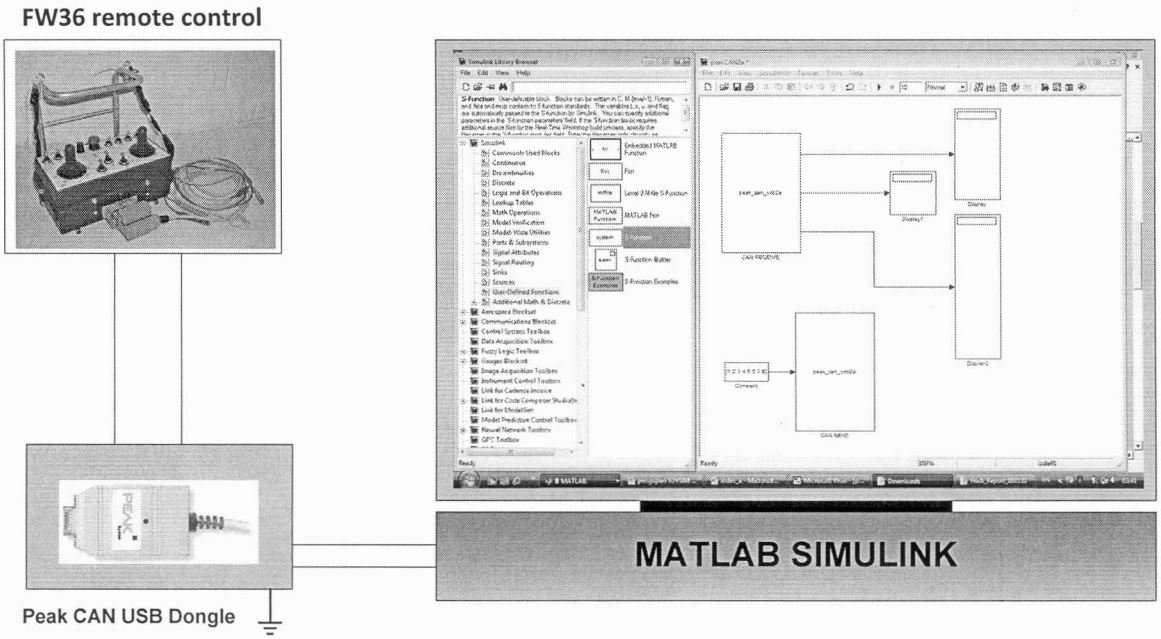
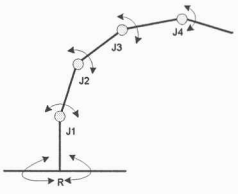


Figure 14. Block Diagram for Function Test



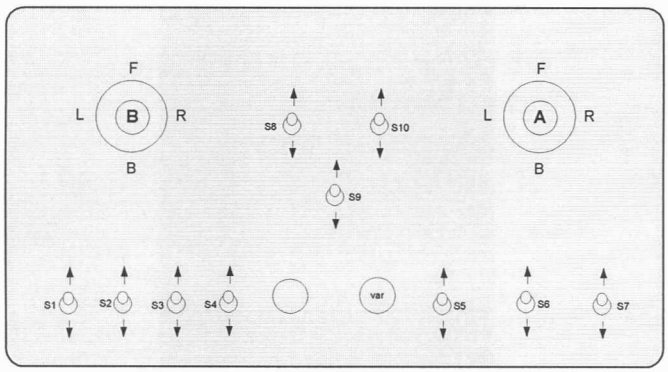


Figure 15. *Fw36 Remote Control* Identification

b. Testing Result

Figure 16 – 19 show test results using a circuit digram as shown in Fig. 14 above.

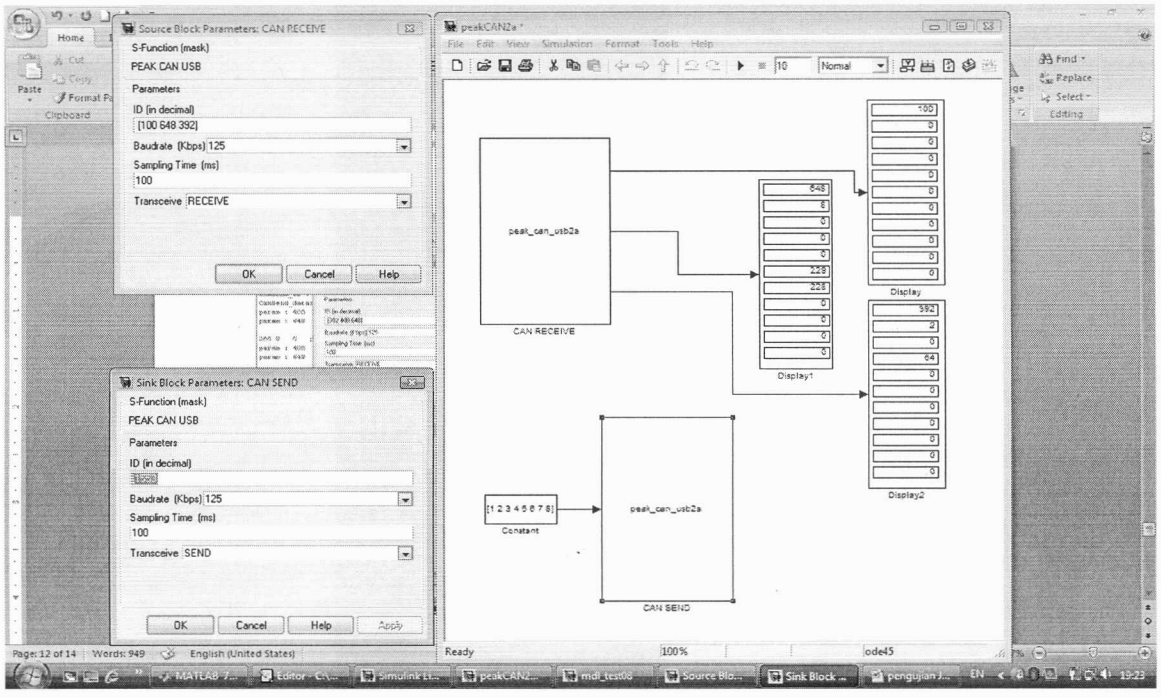


Figure 16. Joystick B (F – L)

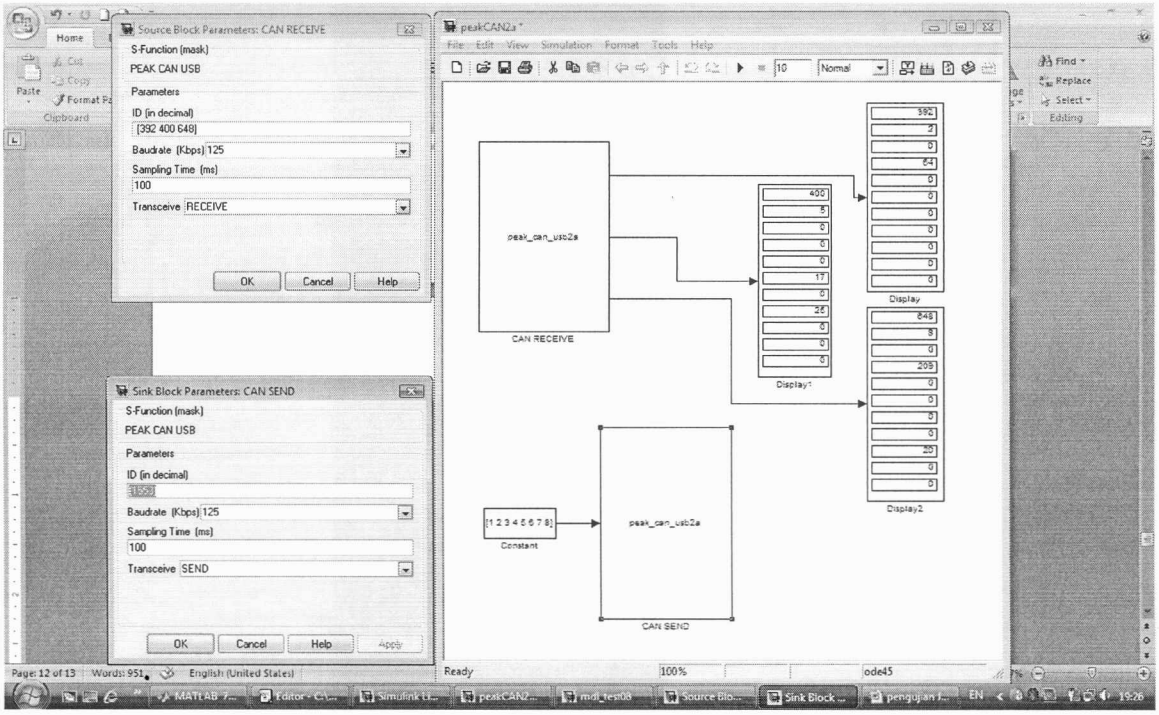


Figure 17. Joystick A (F – L)

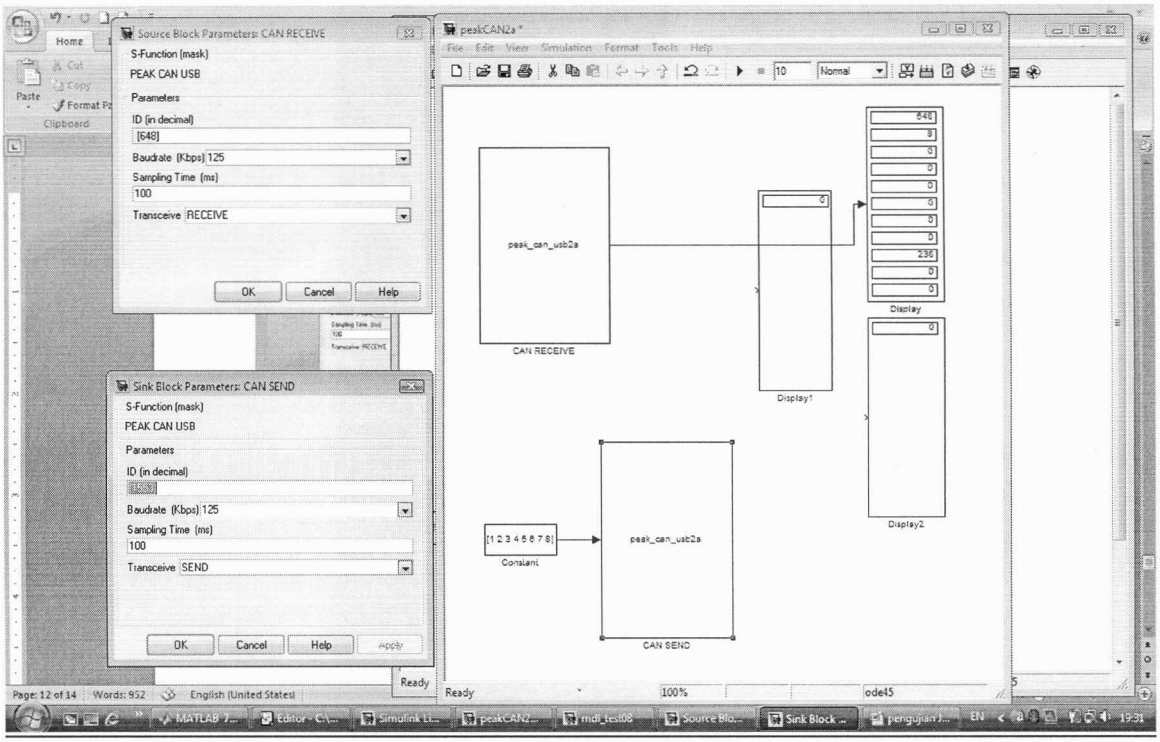


Figure 18. Joystick A (R)

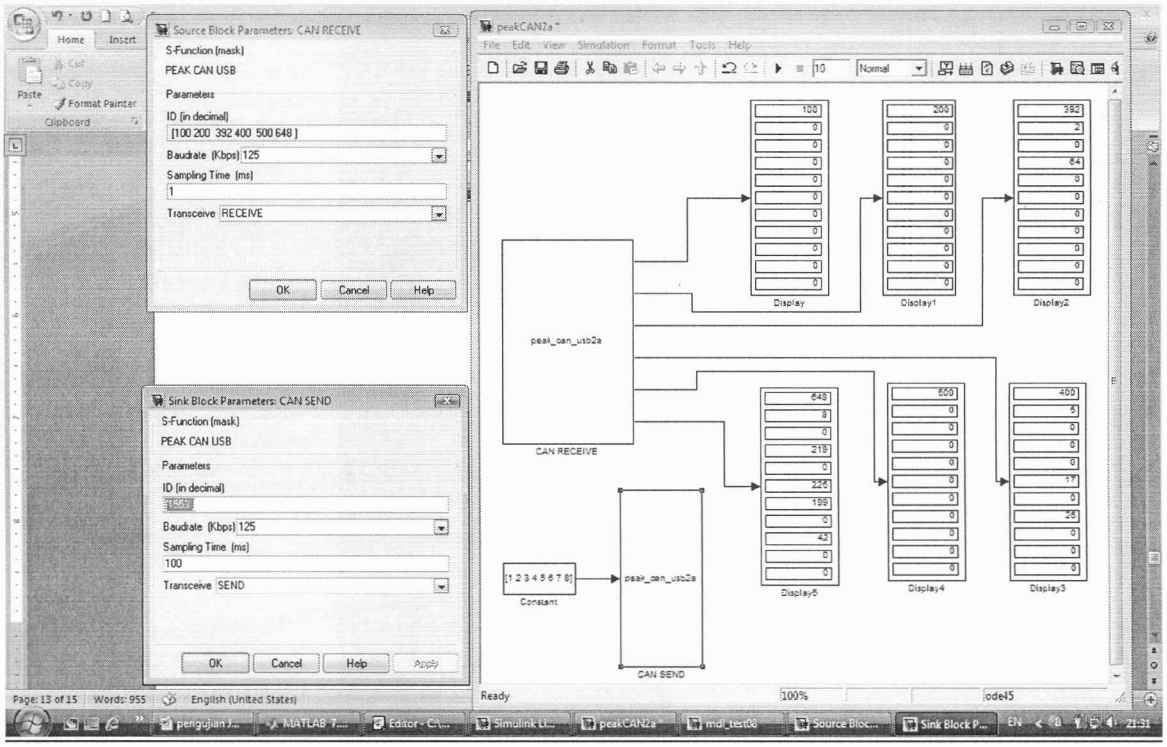


Figure 19. Joystick A (F – L) and Joystick B (F – R)

H. SUMMARY

1. In Matlab/Simulink, the control new functionalities can be implemented easily by more exact adding new modules and connecting them with the present system.
2. Peak CAN USB has three output ID :
    a. 188 H (392),
    b. 190 H (400), and
    c. 288 H (648)