

Programming and communication of two dsPIC30f micro-controller using MATLAB simulink over CAN protocol

¹Ashish Soni, ²Prof. Jagdish Nagar

Department of Electronics & Communication Engineering, Indore. AITR Indore, (M.P.) India
Department of Electronics & Communication Engineering, Indore. AITR Indore, (M.P.) India

Abstract: In this paper, we present the development of a realtime digital system. In this system the basic software tool is MATLAB along with Simulink, Real Time Workshop, and dspic30f Target device, and a C30 compiler. A computer – a host and a target and a compatible data acquisition board is required to provide the interface between the software and the hardware device to be controlled (a custom development system). In this environment, the values of any parameter (control law, inputs) in the Simulink model are easily changed on the host in real time. Measured responses are displayed in real-time. This environment allows for experimentation, and development of controllers for performance comparison of control laws on hardware as shown in the paper
Keywords: MATLAB, Simulink, RTW, dsPIC30f4011, Controller area network.

I. Introduction

When creating project for a new system or device, the use of MATLAB Real Time Workshop and Embedded Coder toolboxes can be a sensible alternative to hand written C code. First, this approach allows easy verification and test of the software before development on the target platform, by embedding the Simulink code in a simulation environment. Additionally, this approach provides platform independence because the Simulink code can be ported (normally without modification) to different platforms simply by use of the appropriate Embedded Coder plug-in. The use of Simulink also aids the production of robust code compared to methods such as hand written C code; this is particularly true for relatively inexperienced C programmers.

We have used this development process to aid in the creation of one significant real time software projects on two different platforms. The first project is communication between two dspic30f4011 micro-controllers using controller area network.

In this project we developed inter setup in MATLAB Simulink, in which we try to filter 3 different parameters of digital engine using CAN protocol which is transmitted by another dsPIC30f4011 micro-controller in MATLAB simulink module, in practical application after flashing code in hardware we have used microchip CAN analyzer hardware for transmitting data from PC to device

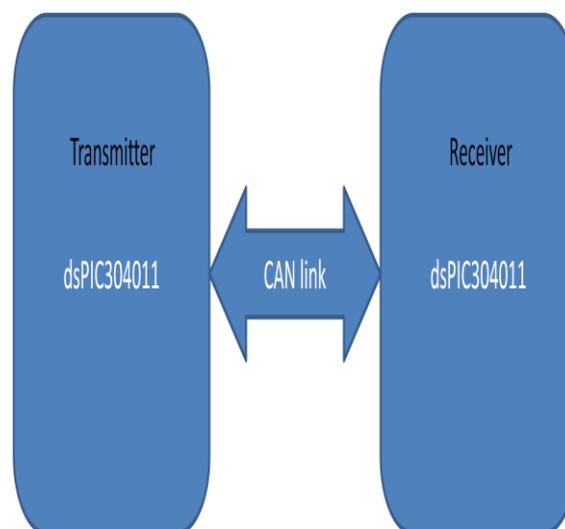


Figure 1 block diagram of transmitter and reviver in MATLAB simulink model base deign

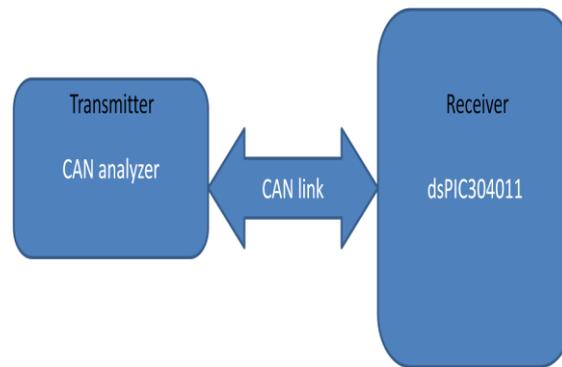


Figure 2 block diagram of transmitter and receiver in practical

Simulink

Simulink, developed by MathWorks, is a data flow graphical programming language tool for modeling, simulation and analyzing multidomain dynamic systems. Its primary interface is a graphical block diagramming tool and a customizable set of block libraries. It offers tight integration with the rest of the MATLAB environment and can either drive MATLAB or be scripted from it. Simulink is widely used in control theory and digital signal processing for multidomain simulation and Model based design [2].

A number of MathWorks and third-party hardware and software products are available for use with Simulink. For example, in simulink we can develop state machines and flow charts with a design environment like stateflow.

Coupled with Simulink Coder, another product from MathWorks, Simulink can automatically generate C source code for real-time implementation of systems called RTW. As the efficiency and flexibility of the code improves, this is becoming more widely used for production systems, in addition to being a popular tool for embedded system design work because of its flexibility and capacity for quick iteration. Embedded Coder creates code efficient enough for use in embedded systems [3].

RTW

Real-Time Workshop is an extension of capabilities of Simulink and MATLAB that automatically generates packages and compiles source code from Simulink models to create real-time software applications [4-5].

- Automatic code generation tailored for different hardware platform.
- A rapid and direct path from system design to implementation
- Seamless integration with Simulink and MATLAB
- A simple graphical user interface
- Extensible make process and an open architecture and

Controller area network

Controller area network (CAN bus) is a bus standard designed to allow microcontrollers and devices to communicate with each other in applications without a host computer. It is a message-based protocol, designed originally for automotive systems [6].

The CAN specification defines the physical and data link layers (layers 1 and 2 in the ISO/OSI reference model). Each CAN frame has seven fields as shown in Figure 1, but we are concerned only with the data length (DL) and the identifier (ID) fields. The DL field is 4 bits wide and specifies the number of data bytes in the data field, from 0 to 8. The ID field can be of two lengths: the standard format is 11-bits, whereas the extended format is 29-bits. It controls both bus arbitration and message addressing, but we are interested only in the former which is described next.

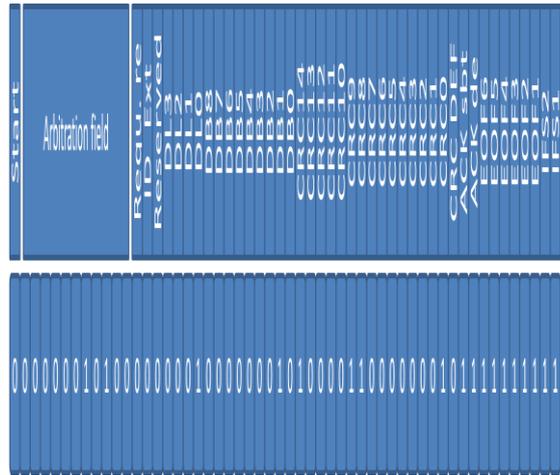


Figure 3 CAN standard frame

Table1:-Standard Frame Bit Description

Field name	Length (bits)	Purpose
Start-of-frame	1	Denotes the start of frame transmission
Identifier (green)	11	A (unique) identifier for the data which also represents the message priority
Remote transmission request (RTR)	1	Dominant (0) (see Remote Frame below)
Identifier extension bit (IDE)	1	Declaring if 11 bit message ID or 29 bit message ID is used. Dominant (0) indicate 11 bit message ID while Recessive (1) indicate 29 bit message.
Reserved bit (r0)	1	Reserved bit (it must be set to dominant (0), but accepted as either dominant or recessive)
Data length code (DLC) (yellow)	4	Number of bytes of data (0–8 bytes)[a]
Data field (red)	0-64	Data to be transmitted (length in bytes dictated by DLC field)
CRC	15	Cyclic redundancy check
CRC delimiter	1	Must be recessive (1)
ACK slot	1	Transmitter sends recessive (1) and any receiver can assert a dominant (0)
ACK delimiter	1	Must be recessive (1)
End-of-frame (EOF)	7	Must be recessive (1)

Decoding algorithm generated by MATLAB simulink:-

```
InData1[1] = C1RX1B2 & 0b1111111100000000;
```

```
InData1[2] = C1RX1B3 & 0b0000000011111111;
```

```
InData1[1] = C1RX1B2>>8;
```

```
InData1[2] = C1RX1B3<<8;
```

```
temp = InData1[1]|InData1[2];
```

```
temp = temp/8;
```

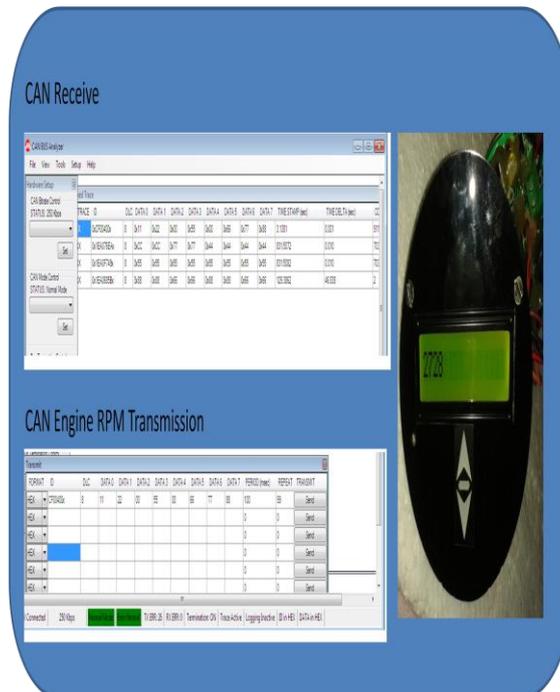
Decoding algorithm is written in interpet subrutin of CAN module

Where, C1RX1B2 and C1RX1B3

16 –bit resistor for holding CAN receiver data

II. Observation and Result

In the figure below we can see that, after flashing C code generated by MATLAB simulink in dsPIC30f4011 micro-controller, we got the data successfully on LCD, which is showing RPM of engine. At left side of window we can see that transmitted EXID CF00400x with data length 8 byte is decode in dspic30f micro-controller to get correct RPM decoding algorithm is shown above.



III. Conclusion:

The custom embedded target enables the user to generate embedded C codes from MATLAB Simulink models and download them into the micro-controllers conveniently. From the figure we can concluded that while using MATLAB simulink we can archive rapid C programming, which can be simulated and tested on MATLAB simulink test bench, RTW from MATLAB simulink is also useful for programmer inexperienced in C.

References

- [1]. J. Roscoe, S. M. Blair and G. M. Burt, "Benchmarking and optimisation of Simulink code using Real-Time Workshop and Embedded Coder for inverter and microgrid control applications", ©2009 IEEE
- [2]. http://www.tutorialspoint.com/matlab/pdf/matlab_simulink.pdf
- [3]. Roberto Saco, Eduardo Pires and Carlos Godfrid, "REAL TIME CONTROLLED LABORATORY PLANT FOR CONTROL EDUCATION", November 6 - 9,2002 IEEE
- [4]. <http://dSPACE.cc.tut.fi/dpub/bitstream/handle/123456789/6857/1/yu.pdf?sequence=3>
- [5]. Dong Zhe, Shi Yuntao, Li Zhijun, Sun Dehui, "Design and implementation of an integrated experimental control platform using rapid control prototyping methods", North China University of Technology, Beijing 100144, July 25-27,2012
- [6]. http://en.wikipedia.org/wiki/CAN_bus