EMBEDDED SYSTEMS TOOLS IN ROBOTICS LABORATORY

Dino Cizmedžic and Jasmin Velagic

Abstract –*With advancing technology ,electrical engineering laboratories are challenged to keep pace.It is especially true for robotics laboratories on universities.The question is what equipment such a laboratory should have in order to be able to implement researches as well as to give students a practical approach to robotics.This paper describes an advantages of Phytec Phycore MPC 555 development board in combination with Metrowerks Codewarrior Development Studio and possibility of code generation from Matlab`s Simulink ,scientific and simalution standard in control engineering .*

Index terms –Embedded system,PowerPC 555, educational equipment,software tools,code generation.

Introduction

With tecnology advances laboratories and curicullum are challenged to keep pace.In this paper we consider an embedded systems solution for research and development use in control systems and robotics laboratory .We were faced with problem of choosing a long term development platform for advanced lab projects as well as providing students with modern educational tool in order to give them better practical approach to modern embedded system.We outlined following criteria for future embedded systems platform:

-Longevity – platform should last at least next 5 years without significant investment in software and hardware.

-Advanced I/O capabilities – systems should be capable for interfacing to A/D,D/A converters, displays,keyboards,sensors and motor driver circuits.

-Advanced processing capabilities – platform should be capable for real time processing of advanced control algorithms.

-Full software support –Wintel platform support and capability of using power of C++ language together with C and assembler.

-RTOS compability –Embedded system should support at least few full documented and available Real Time Operating Systems in order to be applicable in complex application as mobile robot,multiple DOF manipulator or industrial plant controller.

Having been limited with budget we chose Motorola PowerPC 555 development system from Phytec with Motorola`s Metrowerks Codewarrior Integrated Development Enviroment for PowerPC. Unbeknown to us Mathworks offered Emb edded target for Motorola PowerPC 555 toolbox for Simulink,as we also previously considered possibilities of using of Matlab`s Real Time Workshop`s Embedded Coder for generating code Which should have been ported to our embedded system.

Figure 1.   CodeWarrior -PhyCore-Wiggler development tools

Phytec MPC 555 – Codewarrior Platform

Phytec phyCORE MPC-555 is highly functional Single Bord Computer with Motorola PowerPC 555 Microcontroller with 32-bit PowerPC core with 64-bit FPU,40Mhz processor speed,dual CAN 2.0B interface,dual UART,one SPI,two independent TPU units for improved Real Time processing,10 Capture/Compare units with two 16-bit timers,eight PWM output channels,32 analog inputs and several I/O signals.Together with PowerPC`s internal 26 Kb of SRAM and 448 Kb of EEPROM Flash It offers 4MB of external SRAM and 4Mb of external fully programable EEPROM Flash. Board fully supports JTAG/BDM debugging.

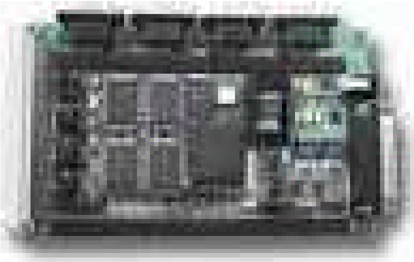

Figure 2. Phytec PhyCore MPC 555 board

Codewarrior IDE offers editor,project manager,class browser,compiler, source and assembly level debuggers,stand alone assembler and linker.Codewarrior debugger supports two debug options for Phytec board:

-Hardware level debugging via BDM interface of PowerPC 555,we are using Macgraigor Wiggler BDM interface for host PC – Phytec board debugging.

-A connection via RS 232 to software debug Monitor kernel (MetroTRK) resident in external or internal Flash on the PhyCore.

CodeWarrior Embedded systems development tool provides powerPC developers with intuitive Graphical-User Interface (GUI), C,C++ - EC++, multiple RTOS support via Kernel aware debug API,full source level debug with building and debugging RAM and ROM images.It offers state of the art features, though not being too complicated for higher grade students for implementing basic control alghoritms (PD,PID...) on PowerPC 555 in control loop with linear plant.

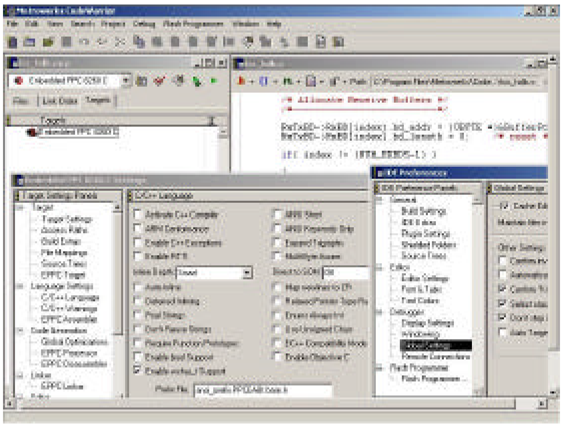

Figure 3. CodeWarrior IDE screen

Matlab`s Embedded Target for MPC 555

he Embedded Target for Motorola MPC555 is an add-on product for use with the Real-Time Workshop Embedded Coder. It provides a complete and unified set of tools for developing embedded applications for the Motorola MPC555 processor.

Used in conjunction with Simulink, Stateflow, and the Real-Time Workshop Embedded Coder, the Embedded Target for Motorola MPC555 lets you:

-Design and model your system and algorithms.

-Compile, download, run and debug generated code on the target hardware, seamlessly integrating with industry-standard compilers and development tools for the MPC555.

The Embedded Target for Motorola MPC555 is an add-on product for use with the Real-Time Workshop Embedded Coder. It provides a complete and unified set of tools for developing embedded applications for the Motorola MPC555 processor.

Used in conjunction with Simulink, Stateflow, and the Real-Time Workshop Embedded Coder, the Embedded Target for Motorola MPC555 lets you:

-Design and model your system and algorithms.

-Compile, download, run and debug generated code on the target hardware, seamlessly integrating with industry-standard compilers and development tools for the MPC555.

The Embedded Target for Motorola MPC555 is an add-on product for use with the Real-Time Workshop Embedded Coder. It provides a complete and unified set of tools for developing embedded applications for the Motorola MPC555 processor.

Used in conjunction with Simulink, Stateflow, and the Real-Time Workshop Embedded Coder, the Embedded Target for Motorola MPC555 lets you:

-Design and model your system and algorithms.

-Compile, download, run and debug generated code on the target hardware, seamlessly integrating with industry-standard compilers and development tools for the MPC555.

The Embedded Target for Motorola MPC555 is an add-on product for use with the Real-Time Workshop Embedded Coder. It provides a complete and unified set of tools for developing embedded applications for the Motorola MPC555 processor.

Used in conjunction with Simulink, Stateflow, and the Real-Time Workshop Embedded Coder, the Embedded Target for Motorola MPC555 lets you:

-Design and model your system and algorithms.

-Compile, download, run and debug generated code on the target hardware, seamlessly integrating with industry-standard compilers and development tools for the MPC555.

The Embedded Target for Motorola MPC555 is an add-on product for use with the Real-Time Workshop Embedded Coder. It provides a complete and unified set of tools for developing embedded applications for the Motorola MPC555 processor. Used in conjunction with Simulink, Stateflow, and the Real-Time Workshop Embedded Coder, the Embedded Target for Motorola MPC555 lets you:

-Design and model your system and algorithms.

-Compile, download, run and debug generated code on the target hardware, seamlessly integrating with industry-standard compilers and development tools for the MPC555.

The Embedded Target for Motorola MPC555 is an add-on product for use with the Real-Time Workshop Embedded Coder. It provides a complete and unified set of tools for developing embedded applications for the Motorola MPC555 processor.

Used in conjunction with Simulink, Stateflow, and the Real-Time Workshop Embedded Coder, the Embedded Target for Motorola MPC555 lets you:

-Design and model your system and algorithms.

-Compile, download, run and debug generated code on the target hardware, seamlessly integrating with industry-standard compilers and development tools for the MPC555.

-Use co-simulation and rapid prototyping techniques to evaluate performance and validate results obtained from generated code running on the target hardware.

-Deploy production code on the target hardware.

The Embedded Target for Motorola MPC555 provides targets that support three application scenarios (using Diab cross-compiler and SingleStep debugger):

-Processor-in-the-loop cosimulation (PIL) target

-Real-time execution and rapid prototyping (RT) target

-Algorithm export (AE) target

The Embedded Target for Motorola MPC555 Library provides device driver blocks that let your applications access on-chip resources. The I/O blocks support the following features of the MPC555: Pulse width modulation (PWM) generation via the Modular Input/Output Subsystem (MIOS) PWM unit.of tools for developing embedded applications for the Motorola MPC555 processor.

Used in conjunction with Simulink, Stateflow, and the Real-Time Workshop Embedded Coder, the Embedded Target for Motorola MPC555 lets you:

-Design and model your system and algorithms.

-Compile, download, run and debug generated code on the target hardware, seamlessly integrating with industry-standard compilers and development tools for the MPC555.

The Embedded Target for Motorola MPC555 is an add-on product for use with the Real-Time Workshop Embedded Coder. It provides a complete and unified set of tools for developing embedded applications for the Motorola MPC555 processor.

Used in conjunction with Simulink, Stateflow, and

the Real-Time Workshop Embedded Coder, the
Embedded Target for Motorola MPC555 lets you:
-Design and model your system and algorithms.
-Compile, download, run and debug generated code
on the target hardware, seamlessly integrating with
industry-standard compilers and development
tools for the MPC555.

The Embedded Target for Motorola MPC555 is an add-on product for use with the Real-Time Workshop Embedded Coder. It provides a complete and unified set of tools for developing embedded applications for the Motorola MPC555 processor. Used in conjunction with Simulink, Stateflow, and the Real-Time Workshop Embedded Coder, the Embedded Target for Motorola MPC555 lets you:

-Design and model your system and algorithms.

-Compile, download, run and debug generated code on the target hardware, seamlessly integrating with industry-standard compilers and development tools for the MPC555.

-Use co-simulation and rapid prototyping techniques to evaluate performance and validate results obtained from generated code running on the target hardware.

-Deploy production code on the target hardware.

The Embedded Target for Motorola MPC555 provides targets that support three application scenarios (using of Diab cross-compiler and SingleStep debugger):

-Processor-in-the-loop cosimulation (PIL) target
-Real-time execution and rapid prototyping (RT) target
-Algorithm export (AE) target

The Embedded Target for Motorola MPC555 Library provides device driver blocks that let your applications access on-chip resources. The I/O blocks support the following features of the MPC555: Pulse width modulation (PWM) generation via the Modular Input/Output Subsystem (MIOS) PWM unit.

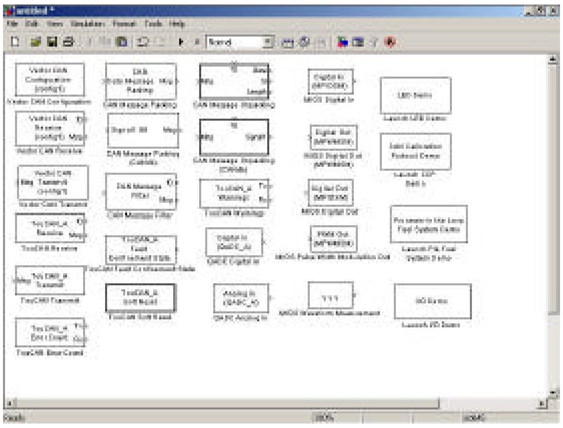

Figure 4. Embedded Target for MPC 555 modules

Analog input via the Queued Analog-to-Digital Converter (QADC64) Digital input and output via the MIOS Digital input via the QADC64 Frequency and pulse width measurement via the MIOS Double Action Submodule (MDASM) Transmit or receive Controller Area Network (CAN) messages via the MPC555 TouCAN modules Utility blocks such as a watchdog timer Device driver blocks support a pass-through option. The pass-through option lets you leave your device driver blocks in your model during simulation. You can then provide a Simulink signal to use in place of the actual device driver.

Unfortunately current version of Embedded target for Motorola PowerPC doesn`t support RT (Real Time target for use with Metrowerks CodeWarrior). We still have option of Processor in Loop cosimulation of closed loop Simulink model for purpose of code validation and analysis. The plant model may contain any Simulink blocks including a combination of continuous-time and discrete-time blocks.

Algorithmexport (AE) target enables you generate code for your controller subsystem and build the code as a stand-alone executable for use on the MPC555. The difference between the AE and the PIL target is that the AE target eliminates all extraneous code (such as serial communications code) used for cosimulation, and also eliminates any real-time interrupts. The AE target generates code only for the basic controller subsystem (e.g. algorithm code). You can then modify or customize this code for your own special purposes.
This option offers us freedom of using RTOS with Simulink generated control algorithms so we have full control in development of complex control system software or agent based software as used in mobile robot control for example.

PIL cosimulation

Processor-in-Loop fault tolerant fuel controller Matlab`s targeting demo consists of few steps considered all RTW options are set for PhyCore target.
After Build subsystem command is chosen from RTW option of fuel controller model, RTW generates appropriate S-function for Simulink

model interfacing for external controller and controller algorithm C code is also being generated.

Analog input via the Queued Analog-to-Digital Converter (QADC64) Digital input and output via the MIOS Digital input via the QADC64 Frequency and pulse width measurement via the MIOS Double Action Submodule (MDASM) Transmit or receive Controller Area Network (CAN) messages via the MPC555 TouCAN modules Utility blocks such as a watchdog timer Device driver blocks support a pass-through option. The pass-through option lets you leave your device driver blocks in your model during simulation. You can then provide a Simulink signal to use in place of the actual device driver

Unfortunately current version of Embedded target for Motorola PowerPC doesn`t support RT (Real Time target for use with Metrowerks CodeWarrior). We still have option of Processor in Loop cosimulation of closed loop Simulink model for purpose of code validation and analysis. The plant model may contain any Simulink blocks including a combination of continuous-time and discrete-time blocks.

Algorithm export (AE) target enables you generate code for your controller subsystem and build the code as a stand-alone executable for use on the MPC555. The difference between the AE and the PIL target is that the AE target eliminates all extraneous code (such as serial communications code) used for cosimulation, and also eliminates any real-time interrupts. The AE target therefore generates code only for the basic controller subsystem (e.g. algorithm code). You can then modify or customize this code for your own special purposes.

This option offers us freedom of using RTOS with Simulink generated control algorithms so we have full control in development of complex control system software or agent based software as used in mobile robot control for example.

PIL cosimulation

Processor-in-Loop fault tolerant fuel controller Matlab`s targeting demo consists of few steps considered all RTW options are set for PhyCore target.

RTW Build subsystem is chosen from RTW options on fuel controller and RTW generates appropriate S-function for Simulink model interfacing for external controller and controller algorithm C code is generated.

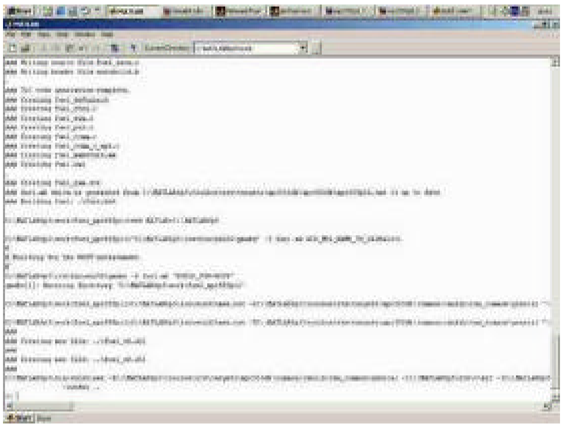

Figure 5. RTW code generation

Simulink automatically opens Codewarrior IDE and generated code is being compiled by Codewarrior.

Figure 6. Codewarrior compiling

After compiled code is being generated and downloaded to PhyCore board Simulink starts cosimulation (non- real time).

Figure 7. PIL cosimulation

Future work

Long term plans for applications with PhyCore - CodeWarrior system include:

-Extension board development with LCD and Keyboard with software drivers.Board would include motor drivers circuits (L298),optocouplers for interfacing with optical encoders ,D/A converters ,and USB controller.

-PXROS RTOS implemented on Phycore board.

-Development of various algorithms (PID, Fuzzy,NN ,LQG) for Real Time control of plants (DC motor,inverted pendulum,tank system,levitating ball).

-Mobile robot,and manipulator control.

-Writing software drivers for interfacing (RS232,USB,CAN,IR) board with external circuits and PC.

Figure 5. RTW code generation

Simulink automatically opens Codewarrior IDE and RTW code is being compiled by Codewarrior.

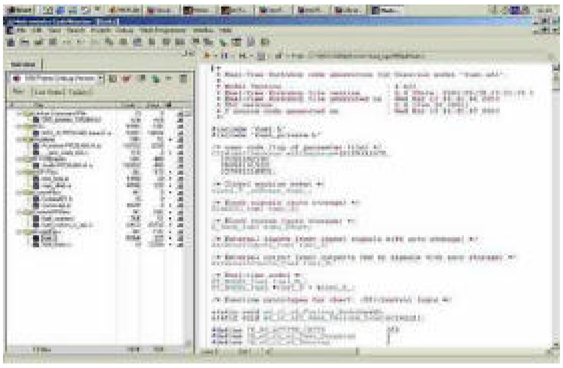

Figure 6. Codewarrior compiling

After compiled code is being generated and downloaded to PhyCore board Simulink starts cosimulation (non- real time).

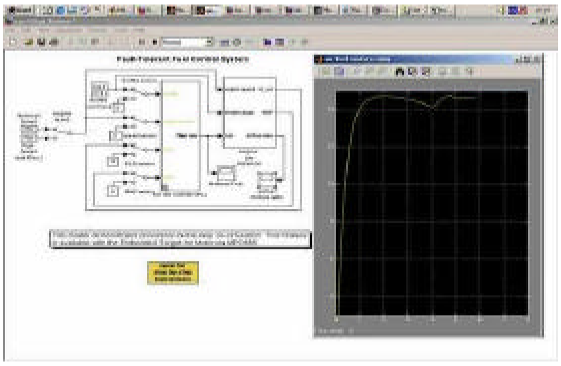

Figure 7. PIL cosimulation

Future work

Long term plans for applications in education and research with PhyCore - CodeWarrior system include:
- Development of extension board with LCD and Keyboard with software drivers.Board would include motor drivers circuits (L298),optocouplers for interfacing with optical encoders ,D/A converters ,and USB controller.
-PXROS RTOS implemented on Phycore board.
-Development of various algorithms (PID, Fuzzy,NN ,LQG) for Real Time control of plants (DC motor,inverted pendulum,tank system,levitating ball).
-Mobile robot,and manipulator control.
-Writing software drivers for interfacing (RS232,USB,CAN,IR) board with external circuits and PC.
-Extensive use of Simulink generated algorithms.

-Development of stand alone controller with PC Graphical User Interface for manipulator control.

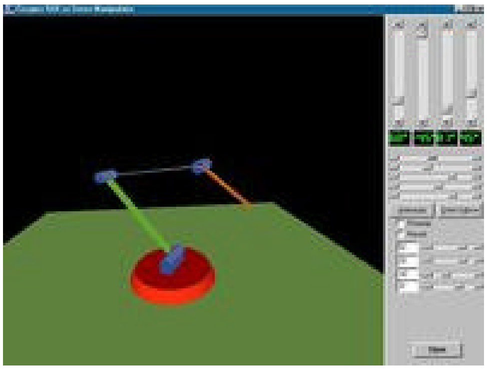

Figure 8. Manipulator Control GUI (Meditronic D.O.O)

REFERENCES

[1] The Mathworks. *"Real Time Workshop for use with Simulink"*, User´s Guide

[2] The Mathworks. *"Embedded Target for Motorola MPC 555"*, User´s Guide

[3] Metrowerks Corporation "Codewarrior development Studio for PowerPC ISA" *Targeting manual*

[4] *Phytec "phyCORE MPC 555 Quickstart Instructions"*

[5] *Aaron Striegel and Diane T. Rover "Enhancing student learning in an introductory embedded systems laboratory" Proc,32nd ASEE/IEEE Frontiers in Education Conference*

Dino Cizmedzic automatika@bih.net.ba
dino.cizmedzic@mittalsteel.com