2004 Programming Quick Reference Guide

Presented by

übergeeks

Team 1028

Table of Contents

Section I

Overview

A. Scope and Purpose

This document is meant to be a quick reference for FIRST teams to use when programming the new PIC-based controller. Please understand this document in its current state is not meant as a full fledged reference, nor as an instruction manual. I hope to expand this into a full reference document, but at present it is merely a quick reference that will (hopefully) cover everything to get you started quick with programming the new EDU-controller.

If you are seeking fuller doucmentation of the controller and related subjects, then please use the list of resources below.

Resources

- Innovation First EDU-Robotics ReadMe
  http://www.innovationfirst.com/FIRSTRobotics/edu-read-me.htm
- Innovation First EDU-Robotics Controller Reference (2004)
  http://innovationfirst.com/FIRSTRobotics/pdfs/EDU_Robot_Controller_Reference_Guide.pdf
- Innovation First EDU-Robotics Programming Guide
  http://www.innovationfirst.com/FIRSTRobotics/pdfs/2004_Programming_Reference_Guide_10-15-2003.pdf
- Innovation First EDU-Robotics 2004 Default Code Reference
  http://www.innovationfirst.com/FIRSTRobotics/pdfs/EDU_Default_Software_Guide_10-15-2003.pdf
- Innvation First Product Page for 2004 EDU-Controller
  http://www.innovationfirst.com/FIRSTRobotics/edu-rc.htm
- MicroChip Product Page for PIC18F8520 (the EDU-Controller Chip)
  http://www.microchip.com/1010/pline/picmicro/category/embctrl/32kbytes/devices/18f8520/index.htm

B. Changelog

October 17, 2003 – Version 0.0.1

- Created document
- Added Basic Table of Contents pertaining to available information on EDU-Controller

C. Additional Comments

Again, I can't stress enough that this document in its current state is **NOT** a full reference. If you are looking for fuller documentation, please look in the resources listed earlier in this section.

Otherwise, enjoy this document! If you make use of this reference and want to offer feedback or you want to add to it, please e-mail me jdmcanally@yahoo.com .

Section II

EDU-Robot Internals & Programming

A. Quick Overview and History

In this section I'll touch on some quick history of FIRST and its EDU-Robotics program. If you've been doing FIRST for a while, you can skip this section and go on to Section B, "Differences Between the 2004 EDU-Robot and the Legacy Edu-Robot".

i. Quick History of FIRST

FIRST inspires in young people, their schools and communities an appreciation of science and technology, and of how mastering these can enrich the lives of all. Corporate America provides economic and professional support to FIRST. Many Fortune 500 companies have significantly helped FIRST grow and are committed to its continued progress. Government organizations such as NASA are key partners in this worthwhile cause. FIRST represents a cooperative team effort by students, teachers, communities, corporations, and our government.

FIRST was created in 1992 with 28 teams in New Hampshire. The event that year was named "Maize Craze" and the competition was held in a high school gym. The FIRST organization has now grown to well over 800 teams with 23 Regional Events and one national Championship Event.

If you'd like more information about FIRST and/or its sponsors and participants, please visit www.usfirst.org .

ii.History of EDU-Robotics

FIRST inagurated the EDU-Robotics program in the 2003 season. This program allowed rookie teams to receive a scaled down yet fully functional robotics kit prior to the season. This kit allowed the rookie teams, many of which who were very unfamiliar with many aspects of building and maintaining a robot (such as the Übergeeks), to practice building, programming, and debugging a simple robot before they tackled a complex full-sized robot in the six-week building period.

In 2004, FIRST revised the EDU-Robotics program to match their revisions to the full-sized robotics kit. They replaced the previous PBASIC-based controller with a new PIC/C-Based controller (These controllers are covered more in-depth in the next section). FIRST also expanded the EDU-Robotics program to offer three distinct packages with varying prices as opposed to the previous one package.

B. Differences Between 2003 & 2004 EDU-Controllers

This section will highlight the main differences between the 2003 and 2004 EDU-Robotics controllers.

i. Hardware[1]

First we'll examine the main differenced in hardware. This section will merely highlight the differences, but attempt to go into enough detail to understand the differences.

a) *Motherboard*

- **CPU[2] –** The 2003 CPU was based on a 50MHz BASIC Stamp processor running at .01 MIPS. The BS2sx (the model number of the 2003 CPU) sports 26 bytes of RAM/variable space with 65 bytes scratchpad space.

  The 2004 CPU on the other hand is based on a 40MHz MicroChip PIC processor running at 10 MIPS. The PIC18F8520 (the model number of the 2004 CPU) sports 1800 bytes (~1.8 Kb) of RAM/variable space + 255 bytes EE[2].
- **Language[3] -** The 2003 CPU was based on the PBASIC language by Parallax (www.parallax.com). The supplied IDE was the PBASIC Editor, a free tool created by Parallax.

  The 2004 CPU is based on the PIC-C language by MicroChip (www.microchip.com). The supplied IDE is the MPLAB-IDE, a commercial tool from MicroChip that is supplied free to FIRST teams or a 60-day trial is available otherwise. Also used is a compiler/libraries "extension" to MPLAB called C18 C Compiler that is also available from MicroChip.
- **Memory –** The 2003 motherboard offered 8 x 2KB banks for holding programs. This was doubled in 2004 by offering 32K single banked space, thus eliminating the need to learn "bank programming" and complicated module procedures.

b) Inputs/Outputs

- **Interrupt Inputs –** Interrupt inputs are a new feature available ONLY on the 2004 EDU-Controller. No references are given in the documentation, so until I can find out more infromation on these inputs this section is under construction.

---

1 See Appendix item 1.1 for basic layout of 2004 EDU-Controller
2 See Appendix item 1.2 for CPU layout
3 To see the order in which a program is executed, see Appendix item 2.1

- **Digital/Analog Inputs –** The 2003 EDU-Controller offered 8 digital inputs and 4 analog inputs. This has been increased in 2004 with a 16 input/output offering that are user definable as either analog or digital inputs.
- **Relay Outputs –** The 2003 EDU-Controller offered 4 Relay outputs for single direction, singular value output (such as switches). The 2004 EDU-Controller offers 6 Mini-solenoid driver outputs (one would assume they're giving small pneumatics this year, even though they aren't listed in the kits?).
- **PWM (Pulse Width Modulation) Outputs –** The 2003 EDU-Controller offered 8 PWM's. These were all standard refresh rate (17 mS). The 2004 EDU-Controller offers 8 PWM's which can use the standard refresh rate (17 mS) or use a new 'fast' refresh rate of 2 mS.
- **Serial** – The 2003 EDU-Controllers Serial ports operated at standard serial (9.6 Kb). The 2004 Controllers operates at TTL Serial (115 Kb). The Program port on the 2004 EDU-Controller operates at RS232 (115 Kb).

c) Other Differences

- **Radio[4] –** The 2003 EDU-Controller offered an onboard 900 Mhz, 40-channel radio. The 2004 EDU-Controller requires a team to use an external RC "hobby" radio.

ii.Programming

Although there are vast hardware differences between the two EDU-Controllers, their programming methodologies are also worlds apart. The 2003 EDU-Controller was based on PBASIC by Parallax (www.parallax.com). The IDE used to create and download the programs to the EDU-Controller was the BASIC Stamp Editor, a free tool provided by Parallax. The PBASIC language itself was primitive, not allowing for subroutines or functions without use of complicated module routines using the programming banks. Its structures were very similar to BASIC mixed with simple Assembly.

The 2004 EDU-Controller is based on PIC-C by MicroChip, a derivative of the C programming language. The IDE used to create programs for the EDU-Controller is MPLAB, a commerical tool provided free (or cheap) to FIRST teams and schools. To download the programs to the EDU-Controller a team must compile them using the C18 Compiler Libraries and then use the InnovationFirst Loader to download them to the robot.

4 For a diagram of the 2004 EDU-Controller R/C control setup, see Appendix 1.3

C. Overview of Files in the Default Code ZIP File

This section will walkthrough the files in the default code provided by InnovationFirst.

PLEASE NOTE

Files denoted with a  MUST NOT be changed.

These files contain VITAL functions that could cause the default code to cease to function if changed.

i. main.c

`main.c` is the file that pulls most of the other files together and contains the `main` function. This function runs through the loops contained in the IFI provided files and the user routines contained in `user_routines` files.

It is advised you leave this file as it is. There shouldn't be any conventional purpose for you to edit this file.

ii. Libraries

a) IFI Libraries

- **ifi_library.h (& ifi_library.lib) –** This header corresponds to `ifi_library.c` (Which I could not find in the ZIP file...?). This file should not be edited.
- **ifi_default.h –** This header contains many important data definitions (relating to `alias.h` and data being sent to the master microprocessor[5]). InnovationFirst recommends that if you decide to edit this file, only ADD to it. Do not change the existing code.
- **ifi_picdefs.h –** This header contains the register definitions for the 18F8520 PIC micro (the next revision of this document will go into detail about the registers). This file should not be edited.
- **ifi_library.h (& ifi_library.lib) –** This header corresponds to `ifi_library.c` (Which I could not find in the ZIP file...?). This file should not be edited.
- **ifi_startup.c –** This source file contains vital startup code and code relating to how you download code to the EDU-Controller. This file should not be edited.
- **ifi_utilities.c and ifi_utilities.h –** This header and source file set contain useful functions for you to call in your programs. This is also where you would want to program any miscelleneous utility subroutines or functions.

b) alias.h

This header file contains many aliases to I/O inputs and outputs on the robot controller, the data sent between these elements, and data sent between the user processor and master processor.

c) Other Libraries

- **printf_lib.c and printf_lib.h –** This header and source file set is a modified version of the standard library for C I/O. These functions can be modified to suit the user's needs.
- **delays.h –** This header file contains cycle-count delay routines for PIC 17Cxxxx and 18Cxxxx PIC chips. I'm not sure why this is included in the source set for an 18Fxxxx chip, but I'm guessing the routines are compatible. This file, although InnovationFirst doesn't specify, probably should not be edited.

iii.User Files

a) `user_routines.c` and `user_routines.h`

This header and source file set are where you would program

5 See Appendix 1.2

any extra user functions and input mappings. You can either modify this file or replace it completely. This file uses the default refresh rate (17 mS).

Please note: `user_routines.c`, `user_routines_fast.c`, and `user_routines_analogex.c` all use user_routines.h.

b) user_routines_fast.c

This source file is where you would program any extra user functions. You can either modify this file or replace it completely. This file uses the fast refresh rate (2 mS).

Please note: `user_routines.c`, `user_routines_fast.c`, and `user_routines_analogex.c` all use user_routines.h.

c) `user_routines_analogex.c`

This source file is where all the input mapping for joysticks and the like are at. You can either modify this file or replace it completely.

Please note: `user_routines.c`, `user_routines_fast.c`, and `user_routines_analogex.c` all use user_routines.h.

C. Programming the EDU-Robot

This section will touch on the methods and tools used to program the EDU-Controller.

i. Getting Started

a) Getting the Tools

There are three methods for obtaining the tools for programming the EDU-Controller.

1. *Wait for the Edu-Kit to arrive* – If you're a rookie team and you're pretty patient, I would recommened just waiting until the Edu-Kit arrives. I'm not sure if they are distributing these kits to every team or just rookies this year, but either way the most sensible suggestion is just to wait.
2. *Download the trial from the MicroChip website* – If you're impatient (like I am), then you can download the 60-day trial of the IDE from the MicroChip site (Link to the file is in the resource list below).
3. *Buy it from MicroChip* – The compiler and libraries are available from MicroChip just for non-school and non-FIRST teams for only $49.95. If you really need it for some reason, then this might be a viable solution.

Resources

- Download 60-Day Trial of MPLAB IDE
  http://www.microchip.com/1010/pline/picmicro/category/embctrl/32kbytes/devices/18f8520/index.htm
- Product Page for EDU-Controller (Link to purchase Compiler is in the table)
  http://www.innovationfirst.com/FIRSTRobotics/edu-rc.htm

b) Knowing the language

I'm sure a lot of you opened up the specifications of the new Controller and FREAKED OUT when you saw it was programmed in C (I know I did, but in a different way I'm sure). Fear not! For I have compiled a decent list of C tutorials to get you familiarized with the structures and methods of C. While you learn this, I'll be hunting down some PIC-specific tutorials, so once you learn C well enough you can beging to program a PIC processor like the ones in the controller.

The resource list below are some of the tutorials I used to learn C and a few others I found on the way. If you know of any great tutorials, e-mail them to me at jdmcanally@yahoo.com.

Resources

- If you're an absolute newbie to programming, you should probably read over this
  http://programmersheaven.com/zone3/cat36/16515.htm
- Main tutorial I used to learn, but not for everyone
  http://programmersheaven.com/zone3/cat36/16574.htm
- Newbie tutorial...Recommended.
  http://programmersheaven.com/zone3/cat36/6628.htm
- By far the best resource on the C programming language, but it costs $40
  http://www.amazon.com/exec/obidos/tg/detail/-/0131103628/ref=ase_learncctoday/102-5040364-3728936?v=glance&s=books

ii.Additional Notes

The next version of this document will be much more detailed in this section. I plan to go in-depth about how to add user routines, which registries are defined and usable in `ifi_picdefs.h`, how to use some of the functions and routines provided by IFI, and more.

Section III

Frequently Asked Questions

Q: What did you use to make this document?
A: I used OpenOffice 1.1 Writer and exported directly to PDF format.

Q: Can I get some more information on your team?
A: Sure. =) Check out www.firstubergeeks.com!

[Yes I realize these are horrible FAQ's, but when I get some FAQ's, I promise I'll put them in here!]

Section IV

Thanks

and Acknowledgements

Big thanks to Chief Delphi and the people on there.
Without you, I would have never known that PWM stood for.

Big thanks for InnovationFirst for providing some excellent documentation.

Big thanks to all of Team 1028 for being my team...yes.

Section V

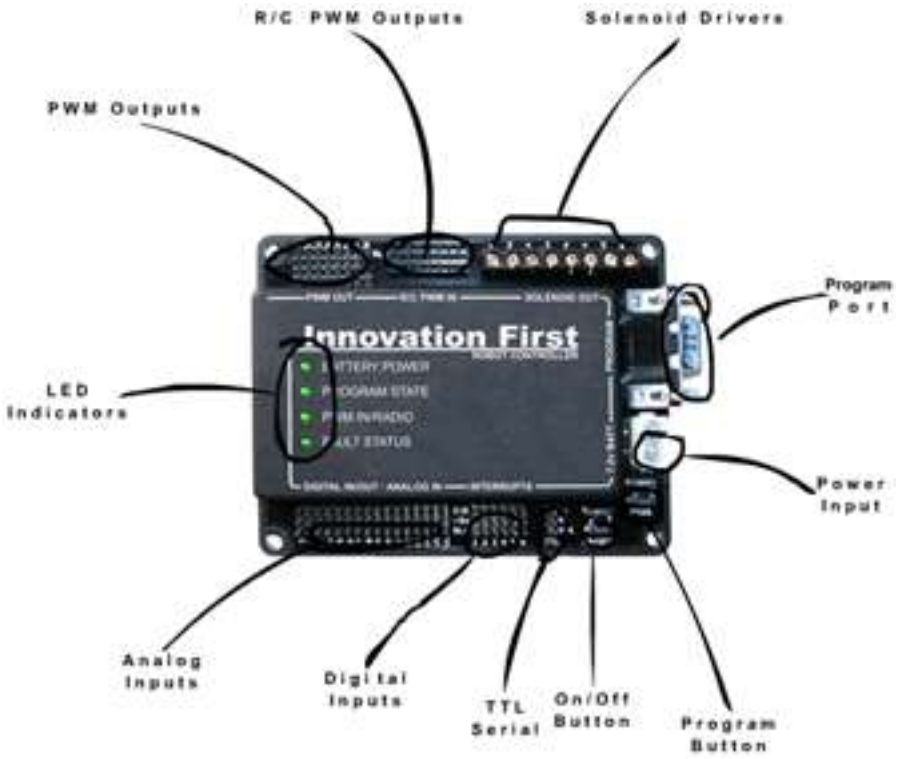
Bibliography

1. 2004 EDU-Robot Controller Reference Guide. October 2003. InnovationFirst, <http://www.innovationfirst.com/FIRSTRobotics/pdfs/EDU-RC-2004_Ref_Guide_10-15-2003.pdf>
2. 2004 FIRST Robotics Programming Reference. October 2003. InnovationFirst, <http://www.innovationfirst.com/FIRSTRobotics/pdfs/2004_Programming_Reference_Guide_10-15-2003.pdf>
3. 2004 FIRST Default Software Guide. October 2003. InnovationFirst, <http://www.innovationfirst.com/FIRSTRobotics/pdfs/EDU_Default_Software_Guide_10-15-2003.pdf>
4. 2004 Edu-Robotics Product Page. 2003. InnovationFirst, <http://www.innovationfirst.com/FIRSTRobotics/edu-rc.htm>
5. 2003 Legacy EDU-Controller Product Page. 2003. InnovationFirst, <http://www.innovationfirst.com/FIRSTRobotics/edu-rc-legacy.htm>
6. MicroChip Product Page for PIC18F8520. 2003. MicroChip, <http://www.microchip.com/1010/pline/picmicro/category/embctrl/32kbytes/devices/18f8520/index.htm>
7. Parallax Stamp Specifications. 2003. Parallax, <http://www.parallax.com/html_pages/tech/faqs/stamp_specs.asp>
8. 2004 Edu-Robotics ReadMe. 2003. InnovationFirst, <http://www.innovationfirst.com/FIRSTRobotics/edu-read-me.htm>
9. 2003 EDU-Controller Reference Guide. InnovationFirst, <http://www.innovationfirst.com/FIRSTRobotics/pdfs/EDU_Robot_Controller_Reference_Guide.pdf>
10. Programmer's Heaven. 1998-2003. 128K-Communications Ltd, <www.programmersheaven.com>

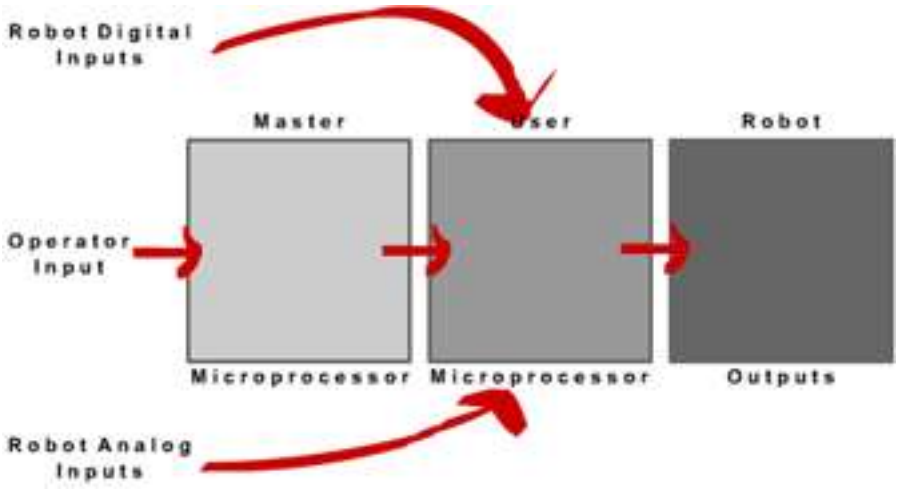
Section VI

Appendix

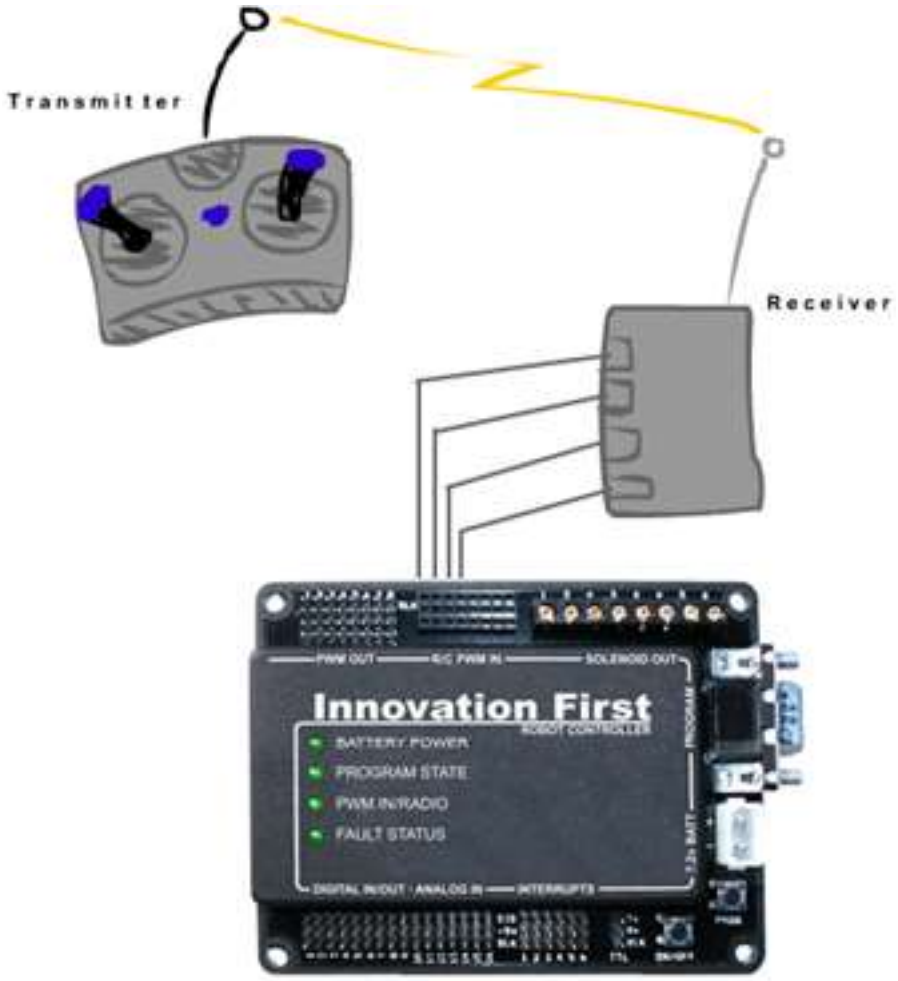
Appendix 1.1 – EDU-Controller Layout



Appendix 1.2 – EDU-Controller CPU's



Appendix 1.3 – R/C Layout for EDU-Robot



Appendix 2.1 – Program Execution