ME 507 Ball and Rail Control Documentation

Authors

Jason Grillo, Aaron Parisi, Trent Peterson

The Ball and Rail control project used the ME 405/507 library provided by John R. Ridgely from the Cal Poly Mechanical Engineering department. The project utilizes a Real Time Operating System (RTOS) to manage the critical timing of many tasks, each designed to execute specific functions simultaneously. The tasks communicate to each other with shared pointers. In short, the documentation includes content from the ME 405/507 library in addition to user source code that was written to implement the state space control algorithm for the Ball and Rail system.

507 Library Information

This documentation describes software designed for the ME405 class at Cal Poly. This software is intended to support the learning of embedded systems programming concepts by undergraduate engineering students. Although primarily intended for education, this software is fully functional in the sense that it can be used for programming projects of a complexity that is found in professional practice. No claim is made that this software is optimal for any such projects, however; see the licenses included for each file for information about the complete and total lack of any warranty associated with this product or any part thereof.

Usage

This software has been developed on Ubuntu Linux sytems using entirely free (as in freedom of choice) development tools which are found in various Linux repositories. However, it can be run on other systems such as Microsoft™ Windows™) computers, and students are encouraged to use whatever environment they prefer when doing development on their own machines.

Makefiles are supplied which automate development on the command line. These makefiles include support for building code projects, downloading to microcontrollers, and building documentation (such as this) using Doxygen. The makefiles and code have settings to support several different processors, and adding settings for more (and, in the case of some drivers, settings for specific target boards) is fairly easy. The user may also choose to work with this software using an integrated development environment such as Eclipse™.

Downloading is usually done using common in-system programmers. A bootloader can be used; we don't prefer bootloaders for class use because we like to use our USB-serial interfaces (or wireless serial interfaces) for debugging and collecting test data from the microcontroller, and the need to switch between programming and debugging use of the serial port makes the test-debug cycle slower and less convenient than the use of separate programmer and debugger do. However, this code is entirely compatible with many bootloaders; this allows upgrading of remote systems from miles away, for example.

Components

Several components are assembled in this software package.

- The FreeRTOS real-time operating system handles preemptive multitasking, scheduling, and inter-task communication. FreeRTOS has a home page at `http://www.freertos.org`.
- FreeRTOS is written in C, not C++, to ensure the widest possible portability. We greatly prefer C++, so a wrapper for FreeRTOS allows user programs to be written almost entirely in C++. The wrapper also extends the debugging support features in FreeRTOS somewhat. The wrapper code was inspired by the Amigo software, by Digital Aggregates Corporation (see `http://www.diag.com/navigation/downloads/Amigo.html`).
- This software supports using the finite state machine software style to organize one's programs. Designing programs using this high-level design technique is an important part of our mechatronics courses.
- There is a class hierarchy to allow very convenient serial output to devices such as wired serial communication ports, radio modems, and SD cards. This library uses features similar to those in the the `iostream` classes of standard C++ (but a much smaller version for use on small microcontrollers).
- A bunch of device drivers have been written to interface various sensors and communication devices.

Targets

There are several circuit boards for which this software has been specifically targeted.

- **ME405 Boards** - These circuit boards are used in the ME405 and ME507 mechatronics courses at Cal Poly. They're designed for motion control and similar applications, thus a pair of high-power VNH series motor drivers from ST Microelectronics as well as various other accessories.
- **Swoop Boards** - These circuits are designed for mechanical measurements (the class, ME410, as well as the application in general). They can hold accelerometers and up to four strain gauge bridge amplifiers and interface with various other sensors.
- **PolyDAQ 1** - This board was designed for the ME236 thermal measurements course at Cal Poly. It has two strain gauge bridges, two thermocouple amplifiers, and two general purpose single-ended voltage inputs. Because it is designed for a thermal measurements course, most measurements are taken at laughably slow speeds, but we'll probably find applications in which higher speeds can be used.
- **PolyDAQ 2** - This upgraded version of the PolyDAQ uses an STM32F405 processor for much higher performance and supports four voltage input channels with a hardware adjustable gain and offset as well as two bridge amplifiers and two thermocouple amplifiers.
- **STM32F4 Discovery** - The PolyDAQ 2 was based on this design. It has a STM32F407 with lots and lots of pins, runs really fast, and comes with an accelerometer and lots of blinky LED's.
- **STM32F4 Nucleo** - This amazingly low-cost board comes with its own STLink2-1 programmer and USB-serial interface. Although not as powerful as a Discovery board, it's hardly a slacker, and its Arduino-compatible header may be handy for many projects.
- **Arduino Mega** - These have been used in some student projects. Their ATmega2560 processors are some of the most powerful in the AVR family and a real pain in the patootey to solder when they need

fixing.

Similar Software

A number of similar software products exist:

- **Arduino** - The very popular embedded programming environment is similar to this software. However, whereas Arduino is intended to insulate the user from many of the dirt-in-your-fingernails basics of embedded programming, this software is intended to help students get their hands dirty and learn those same points.
- **DuinOS** - This is an extension of Arduino with FreeRTOS.
- **Amigo** - A C++ wrapper for FreeRTOS (see above), GPL licensed (thank you) so that its techniques and code could be used here.
- **More** - We're looking for more so that this list can be expanded.

Licenses

The main code in this software package is released under the Lesser GNU Public License, version 2. Other components used with this software have their own licensing terms which can be found through links in the documentation for the files in question; all are free licenses. Some licenses do not restrict commercial use of this software, but most do place certain restrictions on commercial uses.