Recording and Replay System

User's Guide (rev. 02-Mar-17)

Basics

The recording and replay system consists of:

- A robot-specific backend source file
    - *Akagi.c* (or alternatively *RedCastle.c*) implements the control logic for team 3631A's robot (catapult / hang / arcade drive / etc.)
    - *Warspite.c* implements the control logic for team 3631's robot (tank drive and some other apparati)
- *Enterprise.c*, which implements the replay system in RobotC.
- Two separate frontend programs:
    - *Recorder.c* implements teleoperated recording and autonomous replaying.
    - *CompetitionControl.c* implements directly controlled teleop and autonomous replay from a separately-saved recording.

The *Recorder* and *Competition Control* programs are the only two files in this list that can be fully compiled and uploaded to a robot. Attempting to compile any other file separately will give error messages and fail.

Using the Recorder

When using the *Recorder* in Teleop mode, recordings are made automatically: all control input will be automatically saved to a temporary replay, which can be saved at the end of a recording session.

Recording sessions are started automatically by the Recorder—when the program starts, it will display a message on the LCD screen ("Ready to record") and wait for further user input (joystick movement or button presses).

Recording sessions are timed by the recorder, to ensure that replays fit within the allotted autonomous times (15 seconds for normal competition matches, 60 seconds for Programming Skills challenges). When the set time limit is reached, all robot motors will stop, all user controls will be disabled, and a prompt to save will appear on the robot LCD screen (physical and virtual).

The left and right LCD buttons can be used to select the appropriate option, and the center LCD button confirms the action.

The recording session time limit is configurable within the code: at the top of the file, beneath the preprocessor directives and robot port configuration, the line:

```
unsigned int timelimit = 60000;
```

defines the current recording session time limit in units of milliseconds, and can be freely changed.

When placed in Autonomous mode, the recorder will automatically load an appropriate recording and begin replaying it.

Note that switching to Teleop mode using the Virtual Competition Switch (in the RobotC IDE) may cause the robot to begin executing the Autonomous task for a few seconds—double-clicking the button for User Control mode in the IDE prevents this. Nevertheless, make sure you're clear of the robot before switching to Teleop mode.

Multiple "savefiles" are supported by the replay system—the file the Recorder saves to, however, is robot-specific:

- On the 3631A robot (*Akagi* backend), the Recorder will save to and load from the slot currently selected on the program selector dial (labeled 1 / 2 / 3 / Illuminati / Illuminati Skills).
- On the 3631 robot (*Warspite* backend), the Recorder will only save and load from one savefile. The last recorded autonomous will be loaded on robot startup.

Using Competition Control

The *Competition Control* program acts just like a normal robot program. Autonomous mode will load and execute a replay, and Teleop mode will simply receive and interpret user input. Note that user input will not be recorded when using the Competition Control program, and no program-defined time limits are set.

The logic that determines what file is loaded for Autonomous mode is the same that determines what file is saved to using the Recorder (see above)—for the 3631A robot, this is dial-selected, and on the 3631 robot this is the same single savefile.

Manipulating Files on the Robot

The RobotC IDE contains a utility to view and delete files on a connected robot, accessible via "Robot > Advanced Tools > File Management" (the same menu also has the Robot Name utility).

Note that deleting files will require power-cycling the Cortex (saving and loading files does **not** require a power-cycle, however).