

IMPERIAL COLLEGE LONDON

ARES Mars Rover Software

Imperial College Space Society

User Manual and Documentation



Contents

1	Introduction	1
---	--------------	---

List of Figures

List of Tables

1 Introduction

This document is intended to be used as a guide for the installation and use of the ARESRover software and related tools. This software, or more aptly termed, “framework” (at the risk of being accused of spouting buzzwords), is an attempt by the ICSS rover team to create a program that lives on a Mars rover and enables it to perform any task that is typical of a rover. Given this broad goal, a few concrete requirements were realised. This software needs to be able to:

- Handle multiple tasks or “modes” of operation that can run concurrently.
- Provide control of hardware such as sensors and motors to these modes.
- Handle failure gracefully without crashing.
- Provide tools and convenient interfaces for addition of new features.

As of the current (1.0) version, ARESRover consists of two separate components; onboard software and remote software. The onboard software lives on the rover and is structured to meet the above goals. This is the primary component. The remote software is available as a graphical tool for easy communication with, and control of the rover. The onboard software can function independently of the remote software. The onboard software communicates using network sockets using text commands with a specific syntax.

Currently, the onboard software is designed to run on a Raspberry Pi computer, although the software (in a limited form) as well as tests can be run on a laptop or desktop computer. It is written in Python in combination with a few C/C++ extensions. The dependence on Raspberry Pi arises from the use of motor control libraries and the Raspberry Pi camera. This is a weak dependence as the software can easily be extended to support other hardware for motor control, as is planned for future releases. Camera access on desktop/laptop computers is implemented in the current version for testing purposes.

It is important to distinguish between the framework and the implementation that is provided. The implementation makes use of the framework to achieve the functionality required by the ICSS rover for the Olympus Rover trials competition 2019. The components unique to the implementation include the interfaces to the specific hardware used such as an ultrasound sensor and motors accessed via i2c, as well as modes such as the Joystick mode which allows control of the rover using a remote joystick. The framework makes it easy to write such modes and interfaces to hardware. The framework internally handles the starting and stopping of modes using a uniform syntax, and provides appropriate access to the hardware needed by them by taking into consideration other active modes and their hardware use. The framework is the “main” program that runs on the rover (i.e it is the run-time environment that all rover modes operate under).

In the user manual section of this guide, the focus is on the use of the remote software for control of the ICSS implementation. The installation process of the onboard software on a Raspberry Pi is also discussed. This section is relevant if the reader wishes to operate the ICSS rover. The documentation section of this guide is useful to contributors and those who wish to understand the inner workings of the framework. The documentation also covers features specific to the ICSS implementation as a reference and also as an example on how to make use of the framework for writing modes and interfaces to hardware.